

第1章

人工神经网络和模糊信号处理概述

1.1 神经网络

1.2 模糊信号处理

1.3 前景展望

以往的信号处理技术以数学上的严密性为前提，因而无法包含模糊性及主观性等人的因素，近几年随着集成电路技术和计算机技术的飞速发展，在与人的信息连接方面，新型信息处理方式受到了人们的关注，其代表就是神经网络与模糊信号处理。本章将对其进行简要叙述。

1.1 神经网络

神经网络(neural network)和模糊理论(fuzzy theory)是比较新的技术，其应用已经从家电产品扩展到了工业机器等诸多领域。

神经网络是对人脑的信息处理方式进行模拟的产物，与以往的方式相比在原理上存在很大不同。这里将对这项技术出现的背景，特别是有关神经网络的特征等内容予以介绍。

1.1.1 背景

现代科学技术是以对现象进行观察和测定，尽可能严密地用数学公式进行表示和求解为基础的。17世纪的牛顿之前，因为没有再现性的方法，一直把对现象的观察和测定认为不是科学的方法，在用数学公式进行表示和求解之后确立了如图1.1所示的控制理论、系统理论、电路理论、通信理论、网络理论和语言理论等诸多理论。

的确随着科学技术的飞速发展尤其是20世纪60年代人类登上月球之后，世界似乎已经变得越来越小了。就现在的个人计算机而言，具有十几年前大型计算机同样或更高的性能，而体积却只有书本那么大，并且使用对象已经渗透到了个人和家庭。

尽管科学技术发展很快，但是我们也应该认识到科学技术在不同领域的实际应用上还处于较低的水平。

例如声音识别技术，就还没有达到实用水平。虽然周围的喧哗声

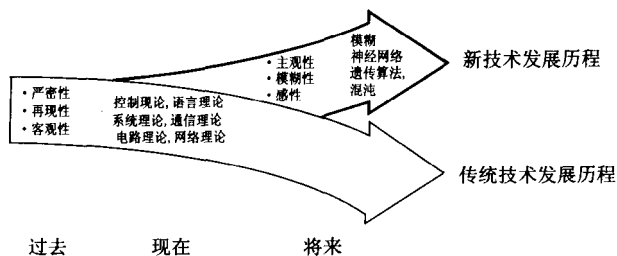


图 1.1 科学技术的发展历程

不能阻碍我们进行谈话，但是实现这种噪音环境下的声音识别仍然是一个难题。还有，我们知道幼儿已经可以在一瞬间认出自己的父母，但是计算机却还不能进行这样的图像识别。

那么，是科学技术不断进步就能够使上述种种难题得以解决，还是科学技术本身就存在某些缺陷呢？

带着这样的疑问，研究者逐渐意识到，以前的科学技术过于追求严密性，而忽视了人类擅长的信息处理方式。也就是，通过经验的积累和不断地学习针对各种各样的问题进行全局考察，弹性对应的方法。目前，人们已经开始利用将这种信息处理方式与严密的科学技术相结合的方法，对涉及到情感、直觉和闪念等单纯利用数值化方法很难解决的问题，进行深入的探讨和研究。

1.1.2 神经网络与模糊理论

为了打破传统科学技术的局限，学者们进行了多方面的具体研究^[1~4]，其中之一就是以人类所具有的主观性为原理构筑而成的模糊理论，有关内容将在下节予以介绍；还有一个，是以世界上高度系统化的人脑信息处理方式为模板，探讨人脑的信息处理方式及其实现，这方面的研究对象称之为神经网络或人工神经网络（artificial neural network）也就是神经元（人工神经元）的网络。

1.1.3 神经网络的兴起

神经网络的兴起是从 20 世纪 80 年代后期开始的。广义的神经网络在这之前就已经从理论研究阶段发展到了应用阶段。这里，根据表 1.1，以通信领域的应用为例予以介绍。

在信号处理中有被称为自适应信号处理¹⁾的一个领域^[5,6]。自适应信号处理的思想是，使系统中的参数具有可变性，从而完成对状况变化的适应性追踪。神经网络的一个很显著的特征，就是优秀的学习能力。自适应可以被广义地认为是学习，所以适应系统可以看作是一个简化了的神经网络。

表 1.1 使用与神经网络相同原理的系统

* 回声消除器
* 自适应预测器
* 未知系统的模型化
* 天线的旁瓣消除器
* 自适应噪声消除器
* 自适应线频谱增强器

比如 在远距离通话时 如图 1.2 所示，语者的声音会经过对方的连接点返回，这是由于连接点中电子线路不匹配引起的。特别是在使用卫星通话的情况下，由于声音到达需要大约 0.3~0.5s 的时间，语者就会一边听着延迟后自己的声音，一边还要说话，这样的通话是非常糟糕的。

这就需要有一个消除回声的装置即回声消除器，其工作原理即是要产生一个与回声等效的信号从而将回声信号从原信号中剔除。不

1) 参看本系列书中的《卡尔曼滤波器与自适应信号处理》一书。

过，不同的通信路线会产生不同程度的回声，因此根据途中线路等的状况还要采取应对时间性变动的措施。由于回声消除器必须吸收这样的时间性变动，可采用如图 1.3 所示的适应形式。

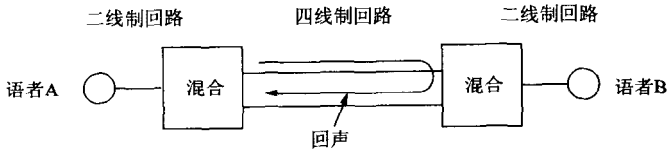


图 1.2 远距离电话线路中回声的产生

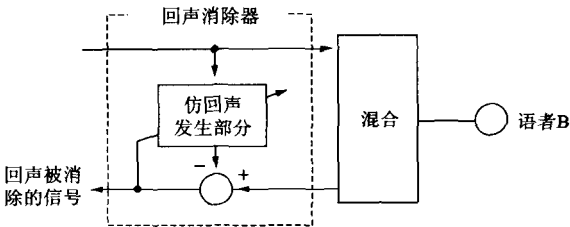


图 1.3 远距离电话线路中的回声消除原理

为使回声成分变为零，仿回声发生部分内的滤波器系数就需要逐次变更。为此，经常用到 **LMS** (Least Mean Square) 算法 (参见 2.2.1 节)，而且，这种算法在神经网络中也被广泛应用。

在声音信号和图像信号的传输中，为了降低通信成本，就要尽可能地减少传输的信息量，所采用的方法是通过过去的信号值对未来值进行某种程度的预测。这样就要用到自适应预测器 (adaptive predictor)。为了能应用于各种信号，必须具备自适应功能的同时还具有较高的通用性。自适应预测器在原理上与回声消除器很相似，都是为了达到实际输入值与预测值之间差值较小为目的而使用 **LMS** 算法。

除了上述介绍的设备之外，还有未知系统的模型化，天线的旁瓣消除器，自适应噪音消除器，自适应线频谱增强器等。总之，神经网络已经被认为是包括自适应信号处理的新型信号处理技术。

1.1.4 神经网络的发展

神经网络中开始受到人们重视的是称为感知器（perceptron）的层次型神经网络。但在当时人工智能（AI: Artificial Intelligence）领域，M. Minsky 等人指出感知器被具有局限性，这使得刚刚兴起的神经网络的发展转入了低谷。

到了 20 世纪 80 年代，人们发现了反向传播（back propagation）在神经网络应用中的优秀学习算法，同时一些具有实际意义的应用相继问世，神经网络一下子又成为了关注的焦点。此外，与物理学比拟的网络的提出和优化问题解法的成功应用，以及计算机处理信息和数据的能力变得更为强大，都是神经网络受到青睐的主要因素。

1.1.5 人类的大脑

为了模拟人脑的信息处理方式，我们首先来了解一下人的大脑。

图 1.4 显示了人脑的结构^[7,8]

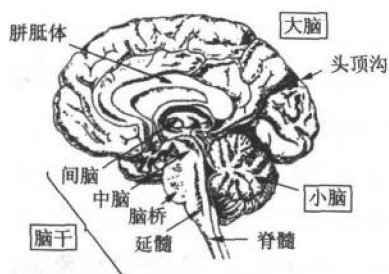


图 1.4 人 脑

脑主要由大脑半球、脑干和小脑组成。大脑半球由大脑皮质构成，低等哺乳类动物中旧皮质和古皮质很多，而人类则有很明显的新皮质结构。大脑皮质的表面有很多沟，据说，如果把它伸展开来，其面积有一张报纸（约 2000 平方厘米）那么大。

如图 1.5 所示，人类大脑皮质的不同部分行使不同的功能。大脑皮质有 6 层结构，大小及形状各异的神经细胞构成了复杂的神经网络。大脑与脑的高级功能如记忆和学习密切相关。脑干中有间脑、中脑、脑桥、延髓等。人的小脑，主要与运动控制及生物功能有关，相当于 3 万台左右普通计算机的处理能力。

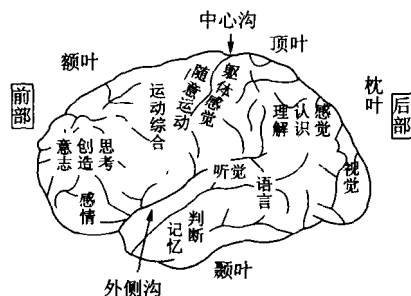


图 1.5 功能与对应的大脑部位

人脑的重量在出生时大约为 400g 左右，成人就达到 1200 ~ 1500g 这不是因为细胞的增加，而是因为神经细胞之间的结合更加紧密的缘故。还有，大脑中氧的消耗速度在身体各内脏器官中是最快的。

1.1.6 神经细胞的模型化

神经细胞有许多种类，但它们的主要功能可以表现为接受从其他神经细胞来的输入，产生与其输入相对应的输出。输入和输出的关系，通常表示为如图 1.6 所示的非线性关系，因为如果用线性关系，就

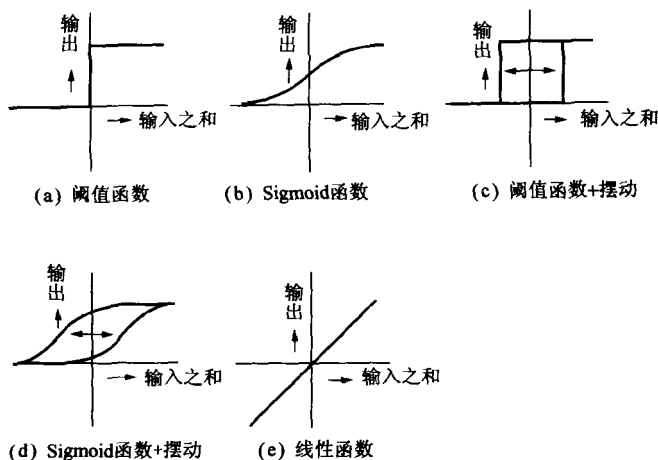


图 1.6 神经细胞的各种模型化

不能显示出多层化的效果。

图 1.6(a) 为最初的形式，是在 McCulloch-Pitts 模型中使用的阈值函数 图 1.6(b) 称为双弯曲函数 (sigmoid function) 应用最为广泛 其特征是容易求得微分值 图 1.6(c) 和 (d) 由于考虑了摆动，具有概率输出的特征 图 1.6(e) 与模糊中用的隶属函数相似。

这些函数有一个共同点，就是输出都被限定在一定范围内，这一点对于网络的稳定是非常有利的。

1.1.7 各种神经网络

若按结构分，神经网络可以分为层次型神经网络和互联型神经网络；若从学习算法角度，可以分为与输入信号对应提供教师信号并且学习输入输出关系的有教师学习，和只根据输入信号进行自组织学习的无教师学习。

层次型神经网络信号的流向，一般为单向。在模式识别及信号符

号化等方面，有教师学习的模型应用比较广泛，特别是反向传播，由于其数学上的明确性和优良的特性而受到了极大的关注。无教师学习的模型则尤其适用于数据的预处理和控制过程当中。

相结合型神经网络信号的流向是双向的，适用于联想存储和优化问题的解法。

表 1.2 给出了各种神经网络及其特征。

注 3.1 本书将在第 2 章和第 3 章中分别详细叙述层次型神经网络和互联型神经网络。

表 1.2 各种神经网络及其特征

网络名称	应 用	注 意 点	特 征
<ul style="list-style-type: none"> 霍普非耳德网络 辅助元 	<ul style="list-style-type: none"> 联想存储模型 	<ul style="list-style-type: none"> 记忆容量比较小 	<ul style="list-style-type: none"> 面向 VLSI
<ul style="list-style-type: none"> 双向联想存储 (BAM) 多向联想存储 (MAM) 	<ul style="list-style-type: none"> 联想存储模型 	<ul style="list-style-type: none"> 记忆容量比较小 	<ul style="list-style-type: none"> 可组合诸多存储项目 面向 VLSI
<ul style="list-style-type: none"> 应用弛豫法的 BAM 和 MAM 	<ul style="list-style-type: none"> 联想存储 情景存储 	<ul style="list-style-type: none"> 需要学习 	<ul style="list-style-type: none"> 记忆容量比较大 可组合诸多存储项目 面向 VLSI
<ul style="list-style-type: none"> 玻尔兹曼机 	<ul style="list-style-type: none"> 组合优化问题 模式识别 	<ul style="list-style-type: none"> 学习时间非常长 	<ul style="list-style-type: none"> 面向 VLSI
<ul style="list-style-type: none"> 平均场理论学习机 (简化的玻尔兹曼机) 	<ul style="list-style-type: none"> 组合优化问题 模式识别 	<ul style="list-style-type: none"> 学习时间比较长 	<ul style="list-style-type: none"> 面向 VLSI
<ul style="list-style-type: none"> 反向传播 	<ul style="list-style-type: none"> 模式识别 	<ul style="list-style-type: none"> 需要很多输入输出数据 	<ul style="list-style-type: none"> 代表性神经网络
<ul style="list-style-type: none"> RBF 网络 (径向基底函数网络) 	<ul style="list-style-type: none"> 模式识别 	<ul style="list-style-type: none"> 需要很多输入输出数据 	<ul style="list-style-type: none"> 学习速度快 对深层次问题不适用
<ul style="list-style-type: none"> 自组织化映射 	<ul style="list-style-type: none"> 输入数据的聚类 特征提取 	<ul style="list-style-type: none"> 需要很多学习 	<ul style="list-style-type: none"> 有生理学依据

续表 1.2

网络名称	应 用	注 意 点	特 征
• 学习向量量子化	• 模式识别 • 向量量子化	• 对深层次问题不适用	• 计算量不大
• 模糊 ART 映射	• 模式识别	• 结构复杂	• 特性很好
• Neocognitorn	• 手写文字识别	• 网络规模大	• 不受文字大小和回转的影响

1.1.8 神经网络的特征

神经网络具有以下所述的显著特征，因此其领域必将有更进一步的扩展。

(1) 学习能力把输入数据和与之对应的输出数据提供给网络，就可以断定其输入输出关系，而且可以针对不同的应用环境，使网络进行学习，通过学习可以使网络变得更加聪明；

(2) 抗故障性一般的系统，即使存在一点点缺陷，也很容易发生较大的故障，或者导致特性大幅度衰减，而对于神经网络，缺陷只造成很细微的特性变化；

(3) 集成性由于神经网络是由比较简单的神经元组合构成的，非常适合于 VLSI 等的集成化¹⁾；

(4) 并行性神经网络主要是进行并行处理，这样就可以很容易地实现高速化处理；

(5) 连贯信息表示神经网络用数个神经元的激活模式表示信息，因而对于过去的人工智能难以表示的连贯信息及联想也都可以表示；

(6) 与脑的类似性由于与生物体的脑具有很高的相似性，随着生理学等的不断进步，也必将会使神经网络有更广阔的发展前景。

1.1.9 神经网络应用举例

表 1.3 列出了一些神经网络的应用例子。

1) 参看本系列书中的《VLSI 与数字信号处理》一书。

表 1.3 神经网络的应用举例

模式识别	控制问题	预测问题	问题解决
声音识别	工厂控制	经济预测	组合优化
文字识别	机器人控制	故障预测	语言处理
图像识别		自然现象预测	

首先在模式识别中，可以完成对声音、文字、图像等的识别。特别是时间延迟神经网络（TDNN: Time Delay Neural Network）在完成对音韵的识别问题上显示出了极强的特性。水果自动分选机和啤酒瓶的缺陷检查装置是应用这项技术的很有趣的例子。

工厂及机器人控制则是在控制问题中的典型例子。

预测问题包括有经济预测、故障预测以及各种自然现象的预测等等。钢铁业和煤气工厂等的异常诊断系统采用的是故障预测，针对被控对象中各部位的压力和温度以及流量等数据进行监视，判断其异常状态和原因。经济预测表现在金融领域的股票价格预测和债券分析等系统上。自然现象的预测主要应用于从地面震动的初期微动开始推断震动的强弱，据此来设计列车的安全运行系统等。

在问题解决方面以组合优化及语言处理等为例，具体地说，通信网络中根据通信量和导线来判定路径，日语假名汉字变换等方面的应用。

综上所述，神经网络的应用可归纳为如下几方面：

预测问题，原因和结果的关系模糊的场合；

模式识别，涉及模糊信息的场合；

不一定非要得到最优解，主要是快速求得与之相近的次优解的场合；

组合的数量非常多，实际求解几乎不可能的场合；

对非线性很高的系统进行控制的场合。

1.2 模糊信号处理

随着美国信息高速公路设想的提出，信息基础设施的建设正实实在

在在地进行着。信息基础设施涉及图像、声音等诸多的数值信号传输，但是人们渐渐发现这些信号很难被掌握。为了从这些繁杂的数值信号中得到人类所能够理解的信号，也就是符号和逻辑，模糊集合（fuzzy sets）及其扩展的概念模糊集合（CFS: Conceptual Fuzzy Sets）就随之产生了。

1.2.1 模糊集合

模糊集合的概念是由美国控制理论专家札德(L. A. Zadeh)教授于 1965 年首先提出来的，它指的是从 0 或 1 的集合向 0 和 1 中间值的扩展，这个概念的提出十分有效地将信息基础设施上的数值信号转变成人们很容易理解的“符号”并明确地表示出了“模糊性程度” [10]

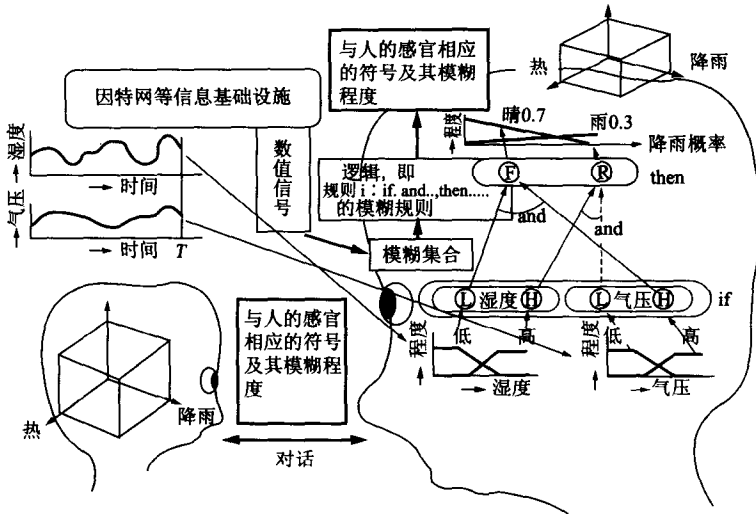


图 1.7 模糊集合与模糊逻辑的一个示例

图 1.7 是一个用模糊集合与模糊逻辑进行简单的天气预报的例子，把湿度及气压的数值信号用函数来表示成低(L)和高(H)等 然后根据

其表示的模糊逻辑 变换成“有无降雨”等与人的亲和性较高的信号。

通过把数值信号转换成“有无降雨”和“热”的程度等与人的感官相应的“符号”和“模糊性程度”之后就可以与其他人对话了。这个例子所表明概念的意义的即以人类可以理解的符号为轴，向其程度表示的多维空间进行变换。

人对数值信号评价的同时，加入通常的状况完成预报。人的预报包含许多使预报有意义的内部状态。人提出的符号包含许多意义要素 就像是冰山的一角 如图 1.8 所示，尽管所看到的部分很小，但其内部却涵盖了复杂的内容。

提出的符号的意义内容，是与当时前后相继的各种关系及其状态相对应的，即图中浮出的冰山所表示的一部分，此外还包含有很多隐含的意义表示部分。在把模糊集合扩展成概念模糊集合^[11]时，如图 1.9 所示，以维特根斯坦(L. J. J. Wittgenstein)的使用学说^[12]为基础，进行了与状态对应的意义内容的表示。下面，从模糊集合的基础到扩展的角度出发，对智能信号处理的有效方法进行简要的介绍。

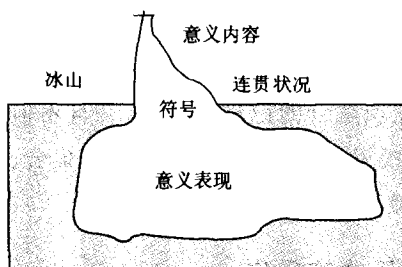


图 1.8 符号和意义的冰山模型

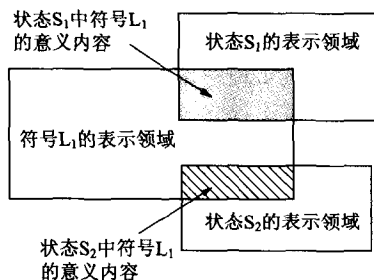


图 1.9 模糊集合的扩展

1.2.2 模糊控制向工业系统的普及

在模糊控制(fuzzy control)^[13,14]中 通过对数值信号的观测 将特

性变动的被控对象的不确定状态变换成与直觉相融合的符号及模糊性程度，从而有效地利用了专家的控制知识。

把图 1.7 所示的模糊信号处理 (fuzzy signal processing) 进行内部构成，就可以把模糊控制系统的应用有效地扩展到许多工业系统中。本书将从模糊控制系统的设计角度，对模糊信号处理的构成方法和稳定性的探讨结合实例予以阐述。

1.2.3 以智能信号处理为目标

随着信息基础设施的不断完善，各种数字化的数值信号也不断出现。要把数值信号变换成人们容易理解的符号和逻辑，首要的问题就是进行信息压缩。在这样的信号处理中，不是 Yes/No 意义的 1/0 记号，而是必须能够与人灵活通信的，模糊性的智能信号处理 (intelligent signal processing)。

因此，要进行模糊信号处理，就出现了根据模糊规则进行信号变换的模糊传感器^[15]。为了在工程上模拟柔性的信号处理，仅用过去的模糊推理形式是不充分的，而是必须要达到自动获取信号变换知识的学习性，和依存于信号融合的传感器联合的信号处理的层次性。

既要实现学习性，又要达到层次性，也就是说，要将现实世界的数值信号变换成人类能够理解，与直觉相融合的记号和命令，并且不但要具有模糊性又可以进行柔性的变换的有效方法，是使用模糊集合、模糊推理和神经网络的模糊信号处理。

1.2.4 模糊信号处理和数值信号的符号化

非线性被控对象的模糊控制指的是在模糊控制中，为了把握被控对象的状态变化，根据与模糊集控制有关的数值信号，转换成类似于表示温度高低等性质的符号和表示其程度的确信度，并使用与设计者的直觉相融合的模糊逻辑，作为命令把结果转换成符号。

这样，将模糊集合中由被控对象观测到的数值，变换成与直觉融

合的性质信号（如输入值的大小、信号的高低）和命令的结果信号（如输出值的大小、高低的信号）来实现某种模糊信号处理，这是对非线性被控对象进行控制的极为有效的手段。

此外，在模糊信号处理领域中，还有与模糊集合并存的模糊测度的研究。模糊集合是用与直觉融合的模糊逻辑，向作为命令的结果符号及其确信度进行变换，而模糊测度则是通过被观测到的数值信号，自上而下被赋予结果的变换，即向着标志其结果的证据程度的方向进行变换^[16]。

以手语单词的识别为例，通过手和指头的变化产生数值信号，对动作部分的证据进行评价，将结果变换为可以识别的单词。但是，存在根据不同状态对同样的手部动作进行判别的层次表示比较困难的问题。

应用过去的模糊集合及模糊逻辑时，尽管根据状态可以表示出要判别的知识，但是对于结果来说就很有可能成为某个单词的确信度的表示，这样就不能充分对动作部分的证据进行评价，一般来说，对于模糊性较大且存在识别结果不明确的情况，需要进行完善。

1.2.5 模糊信号处理和神经网络的融合

人体的信号处理，即使含有模糊性的动作，还是可以通过对部分证据的自下而上处理和根据前后关系及状态确定的自上而下处理进行并行处理，实现动作和话语的识别。这样的自下而上处理和自上而下处理的融合，就需要神经网络的联想记忆。

通过将神经网络和模糊集合及模糊逻辑进行融合，可以实现部分证据的自下而上处理和自上而下处理的结合，从而改善以往的模糊集合中由模糊性的扩大所产生的问题。

而且，使用神经网络，模糊集合和模糊逻辑的学习也变得容易了^[17]。

从控制论(cybernetics)的角度来看，在使用传感器把观测到的无意

识的数值信号转换成有意识的符号，并对这样的信号处理系统进行模型化时，也就是由数值信号到人类所能够理解的符号进行转化的层次模型中的一个重要问题就是要解决隐含的中间状态的符号处理方法。

可以说，在信号处理系统的模型化中，应用神经网络的模糊集合和模糊逻辑，对于解决无意识中隐含的层次性符号的处理方法，是十分有效的。本书还将简要说明这些智能信号处理中的层次性改善方法和自动获得信号变换知识的学习性的改善方法。

1.3 前景展望

为了把人所具有的智力功能即基于人的感官的柔性处理功能进行模型化，软计算正逐渐受到关注。软计算是连同以前的人工智能 (AI: Artificial Intelligence) 技术 把神经网络、模糊系统、遗传算法以及混沌等新型处理进行融合的信息处理技术。它已成为实现新型智能信号处理基础的新技术。

作为新型数字信号处理的硬件技术，通过上述处理的模块化已经实现了高性能化和集成化¹⁾，本书只对软件处理方法上的特征予以介绍。

还要指出一点，作为人的感官处理基础的联想存储和实现新构想的混沌回想方法的统一，可以把信息基础设施上的各种信号自动进行处理从而完成人机对话，这也使得智能 (agent) 的实现成为可能。下面简要介绍一下基于人类感官的智能代理机的并行处理方法和智能自律进化模型。

智能代理机介于人和信息基础设施之间，把数值信号变换成人类易于理解的宏观性符号，而且，还把人的宏观指示反变换成数值信号。

如图 1.10 所示的智能代理机，由于具有构思和学习功能从而可以实现自律 将其编入各种智能系统 就能够组成智能工业系统 (IIS: Inte-

1) 参看本系列书中的《VLSI 与数字信号处理》一书。

lligent Industrial System)。具有能够完成数值信号与人的宏观指示间进行交互式变换的智能工业系统群，对信息设施中的数值信号进行压缩，就能够实现可自律并行处理的多代理机系统(multi-agent System)。

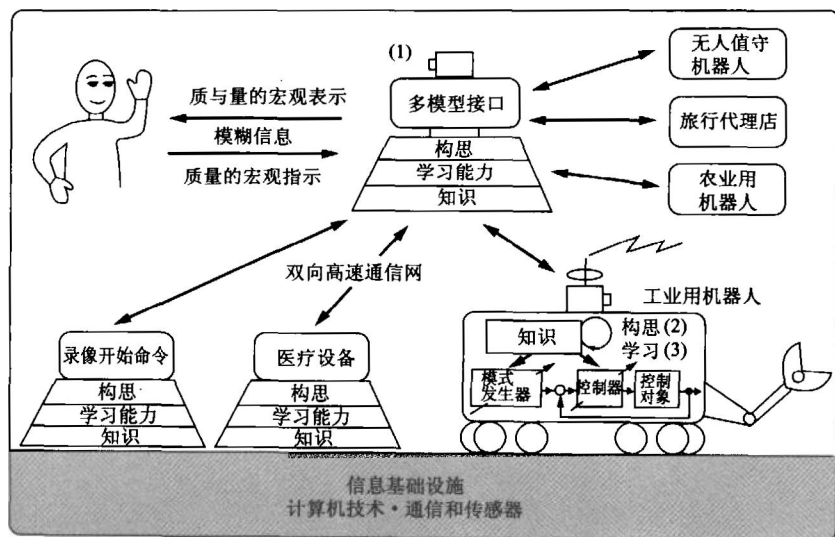


图 1.10 智能代理机和智能工业系统

第2章

层次型神经网络 的信号处理



- 2.1 层次型神经网络概要
- 2.2 反向传播
- 2.3 径向基函数网络
- 2.4 自组织化映射
- 2.5 学习向量量化

神经网络根据网络结构可以分为层次型神经网络和互联型神经网络。在神经网络的实际应用中，层次型神经网络通常应用于模式识别和控制等领域。本章以反向传播学习算法为中心，并对其他的学习算法及其特性等予以介绍。

2.1 层次型神经网络概要

层次型神经网络起源于 20 世纪 60 年代出现的感知器。其后，在 80 年代中期 D. E. Rumelhart 等人发表了称为反向传播(BP)算法的学习算法^[1]同时给出了使用 BP 算法的应用实例^[2,3] 使得其效果受到了人们的普遍关注。

层次型神经网络，由于能够学习输入模式和对应的输出模式之间的关系，在模式识别和控制等方面都有非常广泛的应用。BP 算法的最大特征就是算法明快并且具有十分优异的特性。但是，也存在最优网络结构和中间层神经元的动作尚不明确，学习时间过长等问题。

在改善 BP 算法的同时，其他的学习算法也有了不同程度的发展，包括使用中间层神经元的径向基函数(RBF: Radial Basis Function)的 RBF 网络, T. Kohonen 的自组织映射(self-organizing map)及其改进形式 还有学习向量量化(learning vector quantization)等(参见 2.3 ~ 2.5 节)。

2.2 反向传播

反向传播是神经网络算法中研究得最多并且应用最为广泛的一种学习算法。本节主要介绍算法的原理及推导方法，并对学习的快速化、网络结构的优化方法及应用等予以阐释。

2.2.1 反向传播的基础

BP 算法 是为了层次结构的网络的权值学习 对 LMS 算法一般化的产物。LMS算法是 B. Widrow 等人对优化方法之一的最速下降法 (method of steepest descent) 进行简化而提出的^[4]，该方法使得适应信号处理技术得以快速发展。

这里将介绍最速下降法的原理和 LMS 算法的发展，并在下一节导出 BP 算法。

图 2.1 所示为带有分支延迟线滤波器，在时刻 n 由 M 个输入信号 $x(n-k+1)$ ($k=1, 2, \dots, M$) 输出 $y(n)$ 。滤波器的输入输出关系用下式表示：

$$y(n) = \sum_{k=1}^M w_k(n) x(n-k+1) \quad (2.1)$$

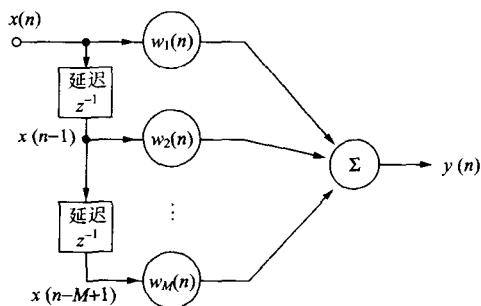


图 2.1 带有分支延迟线滤波器

输入信号为固定的随机时间系列信号，其自相关函数只与时间差有关。滤波器的期望输出为 $d(n)$ 与实际输出 $y(n)$ 的误差表示为

$$e(n) = d(n) - y(n) \quad (2.2)$$

于是，可构成使误差信号的平方均值为最小的滤波器。

$$\epsilon = E[\{d(n) - y(n)\}^2] \quad (2.3)$$

对误差信号进行乘方运算,则有

$$\begin{aligned} e^2(n) &= d^2(n) - 2d(n) \sum_{k=1}^M w_k(n)x(n-k+1) \\ &+ \sum_{k=1}^M \sum_{m=1}^M w_k(n)w_m(n)x(n-k+1)x(n-m+1) \end{aligned} \quad (2.4)$$

由式(2.4)可知,误差信号的乘方是滤波器系数 $w_k(n)$ 的二次函数,对应的图形下凸,极小值即为函数的最小值。由于实际情况有数个滤波器系数,因此该函数为多元函数。

在最速下降法中,误差信号的平方均值是通过求各系数的偏微分来计算各个梯度,如图 2.2 所示要求最小值点。

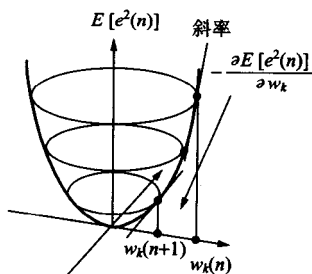


图 2.2 最速下降法原理

对误差信号的平方均值求滤波器系数 $w_k(n)$ 的偏微分 可得

$$\begin{aligned} \frac{\partial E[e^2(n)]}{\partial w_k(n)} &= -2E[d(n)x(n-k+1)] \\ &+ 2E\left[\sum_{m=1}^M w_m(n)x(n-k+1)x(n-m+1)\right] \end{aligned} \quad (2.5)$$

式中右边第一项为

$$\begin{aligned}
(\text{右边第 1 项}) &= -2E\left[\left\{\sum_{m=1}^M w_m(n)x(n-m+1) + e(n)x(n-k+1)\right\}\right] \\
&= -2E\left[\sum_{m=1}^M w_m(n)x(n-m+1)x(n-k+1)\right] \\
&\quad - 2E[e(n)x(n-k+1)] \tag{2.6}
\end{aligned}$$

把式(2.6)代入式(2.5)得到

$$\frac{\partial E[e^2(n)]}{\partial w_k(n)} = -2E[e(n)x(n-k+1)] \tag{2.7}$$

由于与梯度的方向相反，最速下降法的滤波器系数 $w_k(n)$ 的更新量为

$$\begin{aligned}
\Delta w_k(n) &= w_k(n) - w_k(n-1) \\
&= \mu E[e(n)x(n-k+1)] \tag{2.8}
\end{aligned}$$

由式(2.8)可以看出，在最速下降法中，必须求误差信号 $e(n)$ 和输入信号 $x(n-k+1)$ 的平均值，这就存在计算量增大的问题。

Widrow 等人把式(2.8)中的平均值操作进行了省略，设计了下述 LMS 算法：

$$\Delta w_k(n) = \mu e(n)x(n-k+1) \tag{2.9}$$

显然 式(2.9)为

$$(\text{修正量}) = (\text{常数}) \times (\text{误差}) \times (\text{输入}) \tag{2.10}$$

即误差越大，与输入相应的修正量也越大。下面要说明的 BP 算法也是基于同样的原理。

LMS 算法不仅计算量小，而且还有很多其他的优点，因此，在通信均衡器、噪声消除、回声消除等方面被广泛应用，构成了自适应信号处理的基础。

2.2.2 反向传播算法

(1) 反向传播算法的导出 BP 算法基本上用于图 2.3 所示的层次结构的神经网络，相同层的神经元之间没有连接。

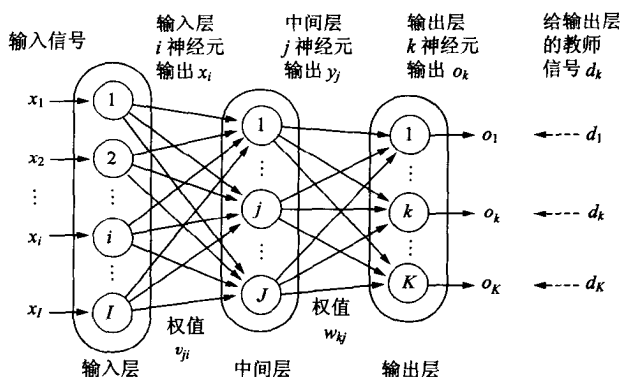


图 2.3 层次结构的神经网络

这里，用图中表示的符号来导出 BP 算法。

将输入信号输入网络，经过中间层传向输出层。输出层的神经元输出和教师信号的平方误差和定义为

$$E = \frac{1}{2} \sum_{k=1}^K (d_k - o_k)^2 \quad (2.11)$$

式中， o_k 表示输出层神经元 k 的输出， net_k 表示输出层神经元 k 的输入和，分别为

$$o_k = f(net_k) \quad (2.12)$$

$$net_k = \sum_{j=1}^J w_{kj} y_j \quad (2.13)$$

应用最小二乘平均原理，先求中间层和输出层间的权值的更新量

$$\begin{aligned} \Delta w_{kj} &= -\eta \frac{\partial E}{\partial w_{kj}} \\ &= -\eta \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial w_{kj}} \\ &= -\eta [-(d_k - o_k)] f'(net_k) y_j \\ &= \eta \delta_k y_j \end{aligned} \quad (2.14)$$

式中， η 为正的常数， δ_k 为输出层神经元 k 的 δ 值。

当 y_j 为中间层神经元 j 的输出 net_j 为中间层神经元 j 的输入和时, 输入层与中间层间的权值更新量有

$$y_j = f(net_j) \quad (\text{中间层神经元 } j \text{ 的输出}) \quad (2.15)$$

$$net_j = \sum_{i=1}^I v_{ji} x_i \quad (\text{中间层神经元 } j \text{ 的输入和}) \quad (2.16)$$

同样, 可求得

$$\begin{aligned} \Delta v_{ji} &= -\eta \frac{\partial E}{\partial v_{ji}} \\ &= -\eta \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial y_j} \frac{\partial y_j}{\partial net_j} \frac{\partial net_j}{\partial v_{ji}} \\ &= -\eta \frac{\partial}{\partial o_k} \left[\frac{1}{2} \sum_{k=1}^K (d_k - o_k)^2 \right] \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial y_j} f'(net_j) x_i \\ &= \eta \sum_{k=1}^K (d_k - o_k) f'(net_k) w_{kj} f'(net_j) x_i \\ &= \eta \sum_{k=1}^K \delta o_k w_{kj} f'(net_j) x_i \\ &= \eta \delta_{yj} x_i \end{aligned} \quad (2.17)$$

式(2.17)取 $k=1 \sim K$ 的和 是因为所有的 net_k 直接依存于 y_j 。 η 为学习常数。

BP 算法中, 除了这样对一个模式逐次进行权值更新的方式外, 还将修正量在全部训练模式上归纳, 同时进行权值更新的批学习方式, 多数情况下逐次更新方式更好一些^[5]

BP 算法的步骤如图 2.4 所示。

(2) 神经元输入输出函数 BP 算法中的神经元输入输出函数应满足的条件为单调增函数, 最常用的是下面的双弯曲函数。

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.18)$$

图 2.5 表示了双弯曲函数及其微分值的大致形状。

那么, 在这里让我们考察一下 BP 的权值更新公式。比如, 式(2.14)和式(2.17)为

1. 网络初始化

权值的初始值用小随机数设定

2. 输入向量的输入(向前)

首先,将输入向量输入给输入层;输入向量向输出层传播;各神经元,求来自前层神经元的附加权值和,由双弯函数决定输出值:

$$(\text{输出值}) = f(\text{输入和}) \quad (1)$$

3. 向输出层输入教师信号

将与输入向量对应的教师向量提供给输出层。

4. 误差逆传播的权值学习

根据

$$(\text{新权值}) = (\text{旧权值}) + (\text{常数}) \times \delta \times (\text{神经元输出}) \quad (2)$$

进行权值更新,式中的(神经元输出)是指权值连接的两个神经元中,来自前面的神经元输出。即通过其权值的值。 δ 按以下求出:

连接输出层的权值学习时,

$$\delta = (\text{输出}) \times [1 - (\text{输出})] \times [(\text{教师信号}) - (\text{神经元输出})] \quad (3)$$

S 函数 $f(x)$ 的微分值

误差

除此以外的权值学习时,

$$\delta = (\text{输出}) \times [1 - (\text{输出})] \times (\text{来自紧接其后层的 } \delta \text{ 的附加权值和}) \quad (4)$$

S 函数 $f(x)$ 的微分值

误差

5. 返回到 2

重复进行 2~4,进行权值的学习。

图 2.4 BP 算法概要

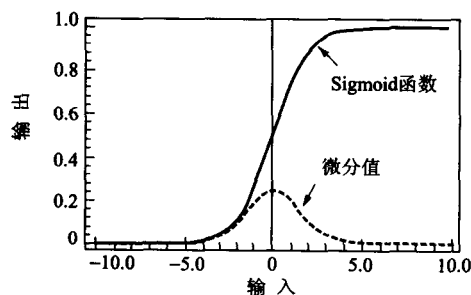


图 2.5 Sigmoid 函数及其微分值

$$\begin{aligned} & (\text{更新量}) = (\text{学习系数}) \times (\text{误差}) \times (\text{Sigmoid函数微分值}) \\ & \times (\text{神经元输出}) \end{aligned} \quad (2.19)$$

实际上, 对 Sigmoid 函数进行微分, 有

$$\begin{aligned} f'(x) &= \frac{1}{(1+e^{-x})^2} e^{-x} \\ &= \frac{1}{1+e^{-x}} \frac{1+e^{-x}-1}{1+e^{-x}} \\ &= f(x)[1-f(x)] \end{aligned}$$

可以看出, 神经元输出 $f(x)$ 在 0 或 1 附近时, 更新量变小, 使得稳定性增强。这样的方法在数理规划中经常用到^[6]

下面讨论对神经元输出范围的影响。式 (2.18) 的神经元输入输出函数中, 神经元输出在 (0,1) 区间内, 但极其接近 0 或 1 的情况却经常发生。这样权值的更新量趋近于 0 学习变得很慢。

那么 输出范围为 (-1,1) 的情形会怎么样呢? 这种情况下, 神经元的激发即使很弱, 也为输出接近 -1 的值, 权值的更新持续进行。因此, 神经元输出范围取为 (-1,1), 学习时间变短。但是, 也有研究指出, 网络特性的通用能力会变得较差^[7]。

(3) 权值的初始值权值和偏置项的初始值, 用较小的随机数设定的情况比较多。值太大, 学习的时间可能很长, 反之, 如果太小, 神经元的输入和很难增大, 也要花很长的学习时间。

当所使用的输入输出函数为奇函数神经元模型时, 初始值设定针对不同的神经元而不同, 理论上在 $[-a/(\text{给神经元输入的权值的数量}), a/(\text{给神经元输入的权值的数量})]$ 范围内比较好^[8] 这里的 a 为 1 位的常数。

2.2.3 学习的快速化

BP 算法有学习时间长缺点, 这里再介绍几个批学习法以外的方法, 可以实现学习的快速化。

(1) 惯性项引入惯性项 (momentum term) 是使得 BP 学习快速

化的方法之一。惯性是借用力学中的惯性概念，不仅考虑当前值，而且对过去的影响也予以考虑的方法。时刻 $t=n$ 求权值修正量时，增加前面时刻 $(n-1)$ 的权值修正量 Δw_{ji}^{n-1} 为：

$$\Delta w_{ji}^n = -\eta \frac{\partial E^{n-1}}{\partial w_{ji}^{n-1}} + \alpha \Delta w_{ji}^{n-1} \quad (2.21)$$

式中 E^{n-1} 为直到时刻 $t=n-1$ 的输出层的平方误差和， α 为惯性项的系数， $0 < \alpha < 1$ 。根据当前值的修正方向（式（2.21）的右边第 1 项）与前一次的权值修正方向（同式右边第 2 项）相异，即符号相反时，求和的修正量的绝对值变小，因此，可以防止过调节。相反这些修正量方向相同时，相加后修正量变为较大的值（绝对值），可以达到学习的快速化。

把评价函数作为从时刻 $t=0$ 开始的平方误差和，通过在权值中假定最新值，可以在理论上推导出惯性项^[9]。

下面，考察一下学习速度、学习常数 η 及惯性项的系数 α 之间的关系。现将

$$g(n) = -\frac{\partial E^n}{\partial w_{ji}^n} \quad (2.22)$$

代入到式（2.21）中，就有

$$\Delta w_{ji}^n = \eta [g(n-1) + \alpha g(n-2) + \cdots + \alpha^{n-1} g(0)] \quad (2.23)$$

$t = N$ 时的权值，由

$$w_{ji}^N - w_{ji}^0 = \sum_{n=1}^N \Delta w_{ji}^n \quad (2.24)$$

可得

$$\begin{aligned} w_{ji}^N - w_{ji}^0 &= \sum_{n=1}^N \eta [g(n-1) + \alpha g(n-2) + \cdots + \alpha^{n-1} g(0)] \\ &= \sum_{n=0}^{N-1} \eta g(n) (1 + \alpha + \alpha^2 + \cdots + \alpha^{N-n-1}) \\ &\cong \frac{\eta}{1-\alpha} \sum_{n=0}^{N-1} g(n) \end{aligned} \quad (2.25)$$

当 $w_{ji}^0 = 0$ ，并且假定评价函数是光滑的，可以看出学习速度大约

与 $\eta/(1-\alpha)$ 成比例。

一般, 由于评价函数为比较复杂的多元函数, 针对不同情况, 惯性项有时对快速化是不起作用的, 也就是说, 惯性项不是在解决所有问题时都有效。

(2) 双弯曲函数微分值的补偿值在 BP 的权值更新中通过双弯曲函数的微分值可以进行学习稳定化。

不过这样的稳定化, 对快速学习为必要的场合反而成为问题。从图 2.5 可以看出, 双弯曲函数的微分值在 $x=0$ 之外都很小, 即使在误差很大时, 权值更新量也常常变得很小。因此, 为了使 $f(x)$ 为非负, 取 $f'(x) \leftarrow f'(x) + (\text{小的正数})$, 经常也用对应于误差的大小给一个偏置成分以增加权值更新量的方法^[10]

(3) **Quickprop** Quickprop^[11]是以 BP 为基础的快速学习算法之一。Quickprop 基于以下两个假定来设计:

每个权值更新量和平方乘误差曲线的关系, 可以近似于抛物线形状(下凸)。

其平方乘误差曲线不受同时被更新的其他权值的影响。

这两个假设很大胆, 与 BP 相比 Quickprop 在很多情况下可以显著提高学习速度。

Quickprop 的权值更新算法, 定义为

$$S(n) = \left. \frac{\partial E}{\partial w_{ij}} \right|_{t=n} \quad (2.26)$$

时, 把 w_{ij}^n 作为在时刻 $t=n$ 的神经元 j 和 i 之间的权值, 可表示为

$$\Delta w_{ij}^n = \frac{S(n)}{S(n-1) - S(n)} \Delta w_{ij}^{n-1} \quad (2.27)$$

可以看出这是一个非常简单的算法。

由式(2.27)可以看出在 $S(n) \approx S(n-1)$ 的情况下, 权值的更新量将变得非常大。因此, 为了防止学习发散, 将新的权值更新量限制在前一次权值更新量的 μ 倍以内。参数 μ 被称为最大成长率 (maximum growth factor)。在解码问题的符号化问题中, μ 的值取 1.75 最为适合。

为了使 Quickprop 更加快速化 在 $S(n)$ 和 $S(n-1)$ 的方向相同时, 也可以使实际的权值更新量为 $(1+\epsilon)$ 倍。

2.2.4 网络结构的优化

应用 BP 算法时, 除了确定 η, α 等学习系数和惯性项系数以外, 还必须确定学习数据的数量和网络结构。如果网络太小, 就会出现能力不足 学习不收敛的问题 相反 如果网络太大 必然使自由度变大 对于学习数据就会产生过剩适应。

如图 2.6 所示 进行 BP 学习时, 一般对应于学习数据, 随着连续学习, 平方误差会变小, 但是对于评价用的未学习数据, 超过某个值时, 反而出现特性变差的现象, 这被称为过学习, 连续学习的一般化能力将下降。

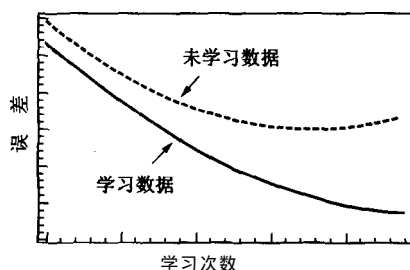


图 2.6 使用反向传播算法时的学习曲线例

这里, 对有关在理论上求网络结构和学习数据数量之间关系的方法, 通过学习确定网络结构的方法予以简要介绍。

(1) 理论上求网络结构的方法 (VC 维数的方法) 使用称为 VC (Vapnik Chervonenkis) 维数的参数求网络结构和学习数据数的关系的方法已为人们所知^[12,13] 这里只简述结论。

VC 维数, 是指是否可以用几种方法把数据分为两部分, 并依此来

表示网络的一般化能力。为了保证一般化能力，输入数据的数量必须大于 VC 维数。

现在考虑输入输出函数为二值的神经网络。输入层和中间层的神经元数分别为 N_i 和 N_h 网络的全部权值数为 N_w 全部神经元数为 N_u 自然对数的底为 e 可以给出 VC 维数，

$$2\left[\frac{N_h}{2}\right]N_i \leq (\text{VC 维数}) \leq 2N_w \log_e N_u \quad (2.28)$$

式中的 $[x]$ 为不超过 x 的最大整数。

下限值是中间层为 1 层的全结合型网络，上限值是适用于任意层数及任意结合样式的网络。当应用 Sigmoid 函数时，VC 维数必须至少取 2 倍以上。实际上输入数据数量为 VC 维数的 10 倍时比较合适^[13]

(2) 由学习确定网络结构的方法 BP 算法的学习，一般中间层神经元的数量越多收敛率越高，反之，数量变少，收敛率将变低。因此，即使对于假设理论上求得的最优网络结构，学习出现困难的情况也很多。

所以，学习初期，中间层神经元的数量设置较多，优先考虑学习的收敛的方法。其后的处理，可以分成在进行学习的同时使网络的规模变小的方法和网络的小型化后再学习重复进行的方法。

学习的同时减小网络规模的方法：把以下有关网络规模的约束项加入到评价函数中，在减小平方误差的同时可以使网络规模变小进行学习。

$$(\text{评价函数}) = (\text{平方误差}) + \lambda (\text{网络规模}) \quad (2.29)$$

式中， λ 为常数 平方误差 E 用下式给出：

$$E = \frac{1}{2} \sum_{k=1}^K (d_k - o_k)^2 \quad (2.30)$$

关于网络规模，提出了以下函数：

$$(\text{网络规模}) = \sum_{i,j} w_{ij}^2 \quad (2.31)$$

$$(\text{网络规模}) = \sum_{i,j} \frac{\left(\frac{w_0}{w_{ij}}\right)^2}{1 + \left(\frac{w_0}{w_{ij}}\right)^2} \quad (2.32)$$

式(2.31)的方法,实际上是对 w_{ij} 偏微分,忽略权值得到的。

$$w_{ij}^{\text{new}} = (1 - \lambda) w_{ij}^{\text{old}} \quad (2.33)$$

这是在适应信号处理领域为了达到系统的稳定而广泛使用的方法。该方法由 Rumelhart 等人提出,特别是在时间序列信号的学习中可以得到良好的特性。

这些方法,对权值消除有效,但不太适用于神经元消除。这是因为,某个神经元的输入或输出的全部权值,如果不为零,就不能消除这个神经元。为了消除神经元,可以把式(2.32)进行扩张,但事实上收敛非常困难。因此,如果考虑把神经元数量减少,以下的方法是比较好的。

网络小型化和再学习重复的方法:首先使学习收敛,然后推断出不要的神经元和权值,重复进行再学习过程。为了推断出不要的神经元用下式求与 k 层 i 神经元连接的全部权值的平方和^[16,17]

$$W_i^k = \sum_j (w_{ij}^{k-1})^2 + \sum_j (w_{ji}^k)^2 \quad (2.34)$$

石边第1项是前层权值的平方和,第2项是向后层连接的权值平方和。这个值越小中间层神经元对网络的贡献越小,优先被消除。消除之前,先把所有权值备份,消除完成后进行再学习。这样的操作学习重复进行。如果不收敛,最后备份的内容作为最终的权值直列收敛。

用以上的方法减少神经元数量后,再消除权值,可以产生更小的网络结构。文献[16]中用与神经元消除相同的想法,进行权值消除。具体地说,从平方和小的权值开始先进行权值消除。应用于镜向对称(mirror symmetry)问题时,一般化能力还可以提高。

另外,由于权值的数量与神经元的数量相比非常多,可以同时为数个权值进行消除,大大提高了效率。

2.2.5 反向传播应用举例

BP 算法是众多神经网络算法中应用最为广泛的算法。这里，以使 BP 成名的 NETtalk 和对时间序列信号（音韵）有较高识别率的时间延迟神经网络（TDNN）为例进行说明。

(1) **NETtalk** NETtalk^[2] 是输入一个英语字串后，将与其中央字串对应的发音符号输出的神经网络。其结构为如图 2.7 所示的简单 3 层神经网络。

在输入层，一个字母用 29 个神经元（26 个拉丁字母以及空格、逗号和句号）表示，另外又取了 7 个字母，全部共需要 $29 \times 7 = 203$ 个神经元。在输出层，用 21 个神经元表示舌的位置以及有声音和无声音等状态，根据这些组合来表现音素。此外，由于用 5 个神经元表示音调，输出层的神经元数就为 26 个。中间层神经元数通过从 0 到 120 个的变化来进行实验。

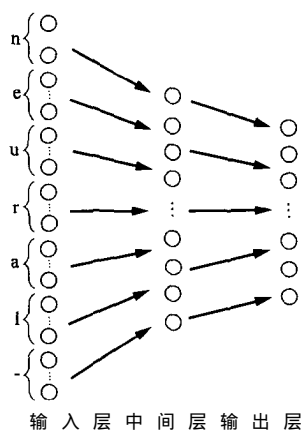


图 2.7 NETtalk 结构

举一个例子，提示一万个语句进行学习时，音素中得到大约 90% 的正确率，对于音调得到大约 100% 的正确率。而且，没有必要用高精度来表示权值，4 位就足够了，中间层的神经元激活的状态处于局部表现和分散表现的中间，从生理学的观点这是一个令人深感兴趣的结果。

(2) 时间延迟神经网络通常的层次型神经网络，应用于像文字识别等的静态模式。但是，涉及如声音信号等时间序列信号时，因为必须吸收时间轴方向的收缩就必须寻求另外的解决办法。在识别声

音时，要求即使发声速度不同也必须能够正确识别。

解决涉及时间序列信号的问题，可以考虑以下方法：

网络中设置反馈，在网络中保留过去信息；

使时间序列信号延迟，在网络中保留过去的信息。

方法 的代表例子是图 2.8 所示的 **Jordan** 网络 (Jordan network)^[18]。该网络通过将输出层的输出看成上下连贯的信息，用返回给输入层这样的形式来实现反馈。

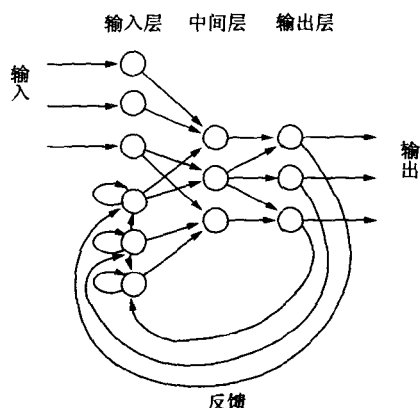


图 2.8 Jordan 网络结构

方法 的典型例子，是提高时间序列信号 (声韵) 识别率的时间延迟神经网络 (TDNN)。图 2.9 是 TDNN 中用到的神经元模型。从结构可以看出图 2.1 中所示的带有分支延迟线滤波器的输入信号，变成了多个的形式。

在带有分支延迟线滤波器中，输出为分支输出的线性和，而在 TDNN 中用非线性函数 (双弯曲函数) 这是最大的区别。

图 2.10 表示的 TDNN 的网络结构，是对三个声韵 / b/、/d/、/g/ 进行识别，整体是由 4 层构成的神经网络。声音信号是以 1 帧 10ms

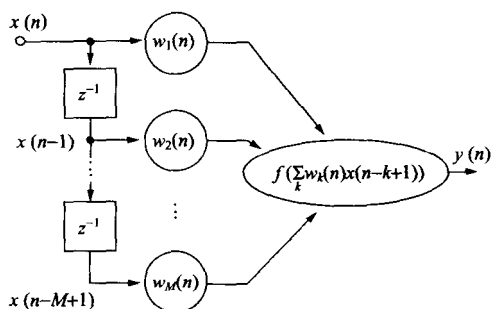


图 2.9 时间延迟神经网络中使用的神经元

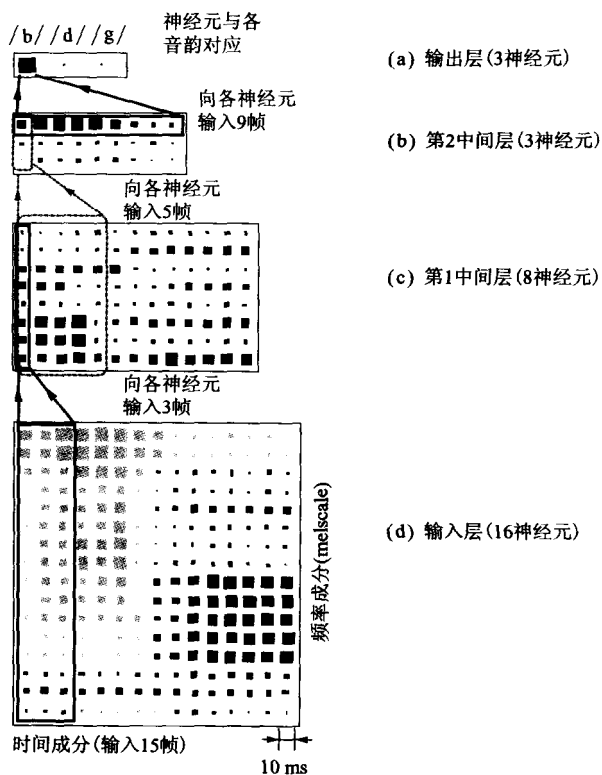


图 2.10 延迟神经网络的结构

在 6kHz 的频域范围内, 用 **Melscale** 方法(与人的听觉适应的频率分割方法)进行频率变换, 把频率带宽分成 16 部分。如图示, 对第 1 中间层的各神经元, 分别输入 1 帧、2 帧、3 帧, 对第 2 中间层的各神经元输入 5 帧。

第 1 中间层之后的神经元是用图 2.9 所示的使用延迟单元的 TDNN 特有的神经元。

对于识别率也进行了细致的实验测定, 比如在进行 / b/、/d/、/g / 的声韵识别时, 在声音识别中广泛采用的 **HMM**(Hidden Markov Model)¹⁾, 即隐式马尔可夫模型为 92.9% 在 TDNN 中得到了 98.8% 的识别率。在其他的声韵识别中, TDNN 方法同样获得了较高识别率。

2.3 径向基函数网络

2.3.1 径向基函数网络基础

下面介绍径向基函数(RBF)网络。RBF 网络的结构如图 2.11 所示, 网络的输出用下式表示:

$$y_k(\mathbf{x}) = \sum_{j=1}^J w_{kj} \phi_j(\mathbf{x} - \mathbf{c}_j), \quad j = 1, 2, \dots, J \quad (2.35)$$

式中 J 为中间层的 RBF 神经元数, w_{kj} 为中间层 j 神经元和输出层 k 神经元之间的权值, \mathbf{c}_j 为第 j 个 RBF 的中心向量, \mathbf{x} 为输入向量。

核函数(kernel function)有各种形式, 经常用到的是下面的高斯型核函数:

$$\phi_j(\mathbf{x} - \mathbf{c}_j) = \exp \left[-\frac{(\mathbf{x} - \mathbf{c}_j)^T (\mathbf{x} - \mathbf{c}_j)}{2\sigma_j^2} \right] \quad (2.36)$$

式中 σ_j 是表示第 j 个 RBF 的幅的参数, T 表示转置。

图 2.12 为大致的形状, 与输入及输出与有单调增加关系的双弯

1) 参看本系列书中的《语音与图像的数字信号处理》一书。

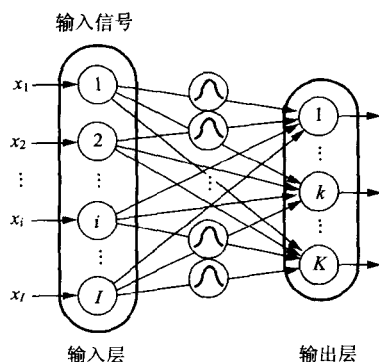


图 2.11 径向基函数网络的结构

曲函数形状的特性不同，中间层的各神经元在输入向量仅与 RBF 的中心向量接近时有较大反应。也就是说，各个 RBF 只对特定的输入有反应。

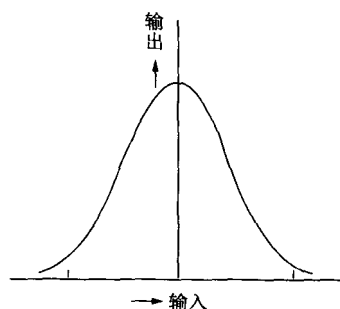


图 2.12 RBF 网络中使用的高斯型核函数

对 RBF 网络提出了各种学习算法，先是中间层，然后是输出层进行学习的两阶段式最常用，在这里，介绍一下有代表性的 J. Moody 和 C. J. Darken 提出的算法^[20]

在 Moody 和 Darken 算法的第 1 阶段中，要决定中间层的 RBF 的

中心位置及表示其宽度的 σ_j 。通常使用聚类方法，这样就成为无教师学习。

第 2 阶段中，中间层和输出层之间的权值，用 LMS 算法等的有教师学习来确定。两段结束之后，还要用有教师学习来对参数进行微调。下面，对各段予以较详细地说明。

第 1 阶段中比较重要的是确定函数 $\phi_j(\cdot)$ 的中心位置，其方法有几种，在这里对应用聚类数固定情况下的聚类方法之一——K 平均法 (K-means) 进行介绍。聚类数取 J 数据数取 I 则其算法为：

第 1 步 聚类中心向量的初始化 确定 J 个初始聚类中心向量 $c_{10}, c_{20}, \dots, c_{J0}$ 。数据集中，常把最初 J 个数据作为初期聚类中心向量。

第 2 步：数据输入及对应聚类选择：输入数据 x_i 而且选择最近的聚类 j^* 。

$$c_{j^*} = \arg \min_j \|x_i - c_j\| \quad (2.37)$$

将其对全部数据 ($i=1, 2, \dots, I$) 进行。

第 3 步：聚类中心向量的更新：聚类中心向量按下式更新：

$$c_j^{\text{new}} = \frac{1}{N_j} \sum_{x \in \text{cluster}(j)} x \quad (2.38)$$

式中 N_j 是属于 J 聚类的数据数。将其对全部数据 ($j=1, 2, \dots, I$) 进行。

第 4 步：判定：如果全部聚类中心向量没有变化，进行第 2 阶段，否则返回第 2 步。

第 2 阶段中，中间层和输出层间的权值 w_{kj} 用有教师学习确定，其应用 LMS 算法的方法是：

第 5 步：中间层和输出层间的权值的初始化：中间层和输出层之间的权值 w_{kj} 用比较小的随机数设定。

第 6 步 网络输出的计算 输入数据 由式 (2.35) 和式 (2.36) 求网络的输出。式 (2.36) 中关于 RBF 的宽度 σ_j ，经常用到下式的聚类中心和属于其聚类的数据之间的平均距离：

$$\sigma_j^2 = \frac{1}{N_j} \sum_{x \in \text{cluster}(j)} (x - c_j)^T (x - c_j) \quad (2.39)$$

第 7 步:权值的更新: 首先求各输出神经元中的误差。

$$e_k = d_k - y_k(x) \quad (2.40)$$

式中, d_k 是输出神经元 k 的希望输出。然后, 按下式更新权值:

$$w_{kj}^{\text{new}} = w_{kj}^{\text{old}} + \eta e_k \phi_j(x - c_j) \quad (2.41)$$

式中, η 是学习常数。

第 8 步:判定:如果满足事先确定的终止条件, 结束。否则返回第 6 步

2.3.2 RBF 网络和 BP 网络的比较

RBF 网络也和 BP 网络一样, 对任何非线性连续映射能够用任意精度近似。但是, 还存在以下几方面的区别:

(1) 数学解释在数学上 BP 网络为输入输出函数的概率近似 (stochastic approximation) 与此相对在 RBF 中是进行判定边界面的近似 (curve fitting)。在 BP 网络中, 各神经元的输入由于为前层输出的 1 次结合, 其判定边界面为直线形状的组合。

由于 RBF 使用径向基函数, 呈圆形或椭圆形, 因此, 对判定边界面的形状有一定程度了解的情况下, 可以把两种网络分开使用。

(2) 中间层神经元的作用 BP 网络是为了使输出层的误差为最小而进行的权值学习。各神经元具有同样的输入输出特性, 但中间层的各神经元的作用却不那么明确。

RBF 是基于将模式识别等问题用非线性向高维投影, 从而容易进行分离的考虑。中间层的各神经元具有不同的输入输出特性, 因而各自作用是明确的。

(3) 学习速度 RBF 的学习速度更快。这是因为, 很多 RBF 的学习算法能够分成两段, 各自都能快速化, 而 BP 网络必须同时学习全部权值。而且, 由于 BP 网络是多层结构, 如果前层的权值不确定的话, 后面层的权值也不确定, 所以, 学习时间就有很大的差异。

2.3.3 RBF 网络

前面介绍的 RBF 网络是基本型,此外还有以下几种形式:

(1) 高斯型核函数的一般化式(2.36)的高斯型核函数 对任意个轴的半径都相同,函数的形状为圆形。在更一般化的高斯型核函数中,考虑到各个轴的不同半径,函数的形状为椭圆形,这样就可以更细地近似,但是,计算量也要增大。文献[22]中提出了用格拉姆-施密特(Gram-Schmidt)正交化法的方法。

(2) 中间层神经元数的自动确定 RBF 网络也和 BP 网络一样,中间层神经元数很大程度上支配着网络的特性。为了自动确定中间层神经元数,可以使用将评价函数设为平方误差项和网络规模项的和的方法^[23]。还有根据需要增加中间层神经元的方法^[24] 还可以采取最初准备较多的中间层神经元,然后边融合边减少等方法^[22]。

(3) 小波神经网络目前,作为新的波形解析方法,小波解析(wavelet analysis)备受瞩目。所谓小波,是取微小波纹之意,对称为基础小波的波形 $\psi(x)$ 施以平行移动和伸缩的操作而获得的函数序列

$$f(x) = \psi\left(\frac{x-b}{a}\right) \quad (2.42)$$

对其进行波形解析,使用这一方法的小波神经网络也出现了。它与 RBF 网络在数学上不同,但网络结构却类似^[26]。可以预见,引进小波解析思想的神经网络将会不断涌现。

2.4 自组织化映射

2.4.1 无教师学习

到目前为止介绍的算法,是以与输入数据对应的教师数据为必要的有教师学习算法。然而,在实际应用中,很多情况下没有提供教师

数据，这时，就要用到无教师学习。

无教师学习的特征有以下几点：

聚类：适用于将数据集分类成为几组的情况；

数据自动提取特征：能够自动提取数据集中隐藏的特征；

类似性和典型性的表现：用无教师学习时，输出层神经元表现出固有特征的情况很多。因此，根据其输出值可以把对应于各自的特征的类似性和典型性等表现出来；

输入数据间的相互关系的表示：输入数据的维数比较高时，输入数据之间的相互关系就很难把握。在无教师学习中，可以将输入数据映射成任意维，并且可以将数据之间的相互关系进行可视化；

信息压缩：与输入层神经元数相比，通过减少输出层神经元数，可以消除冗余信息，进行信息压缩。

2.4.2 自组织化映射

下面说明作为无教师学习算法代表的 Kohonen 的自组织化映射算法^[27~29]。

Kohonen 的自组织化映射算法，能够自动找出输入数据之间的相似度，将相似的输入在网络上就近配置。因此可以构成对输入数据有选择地予以反应的网络。

自组织化映射 如图 2.13 所示，基本上为输入层和映射层的双层结构。同一层的神经元之间没有结合。

Kohonen 的自组织化映射的学习算法步骤归纳如下：

第 1 步：网络初始化：输入层和映射层之间的权值初始值，用较小的随机数设定。

第 2 步 输入向量的输入 首先 把输入向量

$$\mathbf{x} = [x_1, x_2, x_3, \dots, x_n]^T \quad (2.43)$$

输入给输入层。

第 3 步：计算映射层的权值向量和输入向量的距离：在映射层，计

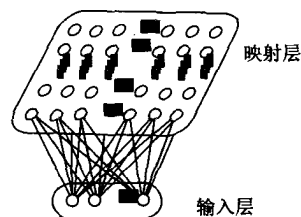


图 2.13 Kohonen 自组织化映射的结构

算各神经元的权值向量和输入向量的距离。这里，映射层的第 j 个神经元和输入向量的距离，按下式给出：

$$d_j = \sum_{i=1}^n (x_i - w_{ji})^2 \quad (2.44)$$

式中 w_{ji} 为输入层的 i 神经元和映射层的 j 神经元之间的权值。

第 4 步：与权值向量的距离为最小的神经元的选择 d_j 为最小，即选择使输入向量和权值向量的距离为最小的神经元。把其称为胜出神经元 (winner neuron)，记为 j^* 。

第 5 步：权值的学习：胜出神经元和位于其邻域的神经元的权值，按下式更新：

$$\Delta w_{ji} = \eta h(j, j^*) (x_i - w_{ji}) \quad (2.45)$$

式中 η 为正的学习常数 $h(j, j^*)$ 称为邻域函数 (neighborhood function)，用下式表示：

$$h(j, j^*) = \exp\left(-\frac{|j - j^*|^2}{\sigma^2}\right) \quad (2.46)$$

上式的 σ 随着学习的进行而减小。因此， $h(j, j^*)$ 的范围，如图 2.14 所示，学习初期很宽，随着学习的进行而变窄。也就是说，随着学习的进行从粗调整向微调变化。这样，邻域函数 $h(j, j^*)$ 可以起到有效产生映射的作用。

第 6 步 返回第 2 步 重复第 2 步到第 5 步 进行权值的学习。

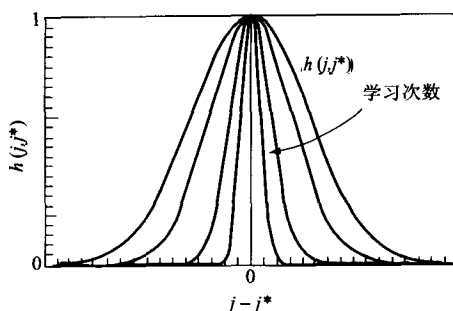


图 2.14 邻域函数的范围变化

在自组织化映射中，由式 (2.45) 可见，胜出神经元和其附近的神经元全部接近当时的输入向量。学习初期，根据邻域函数 $h(j, j^*)$ ，在附近有很多神经元，形成粗略的映射。随着学习的进行， $h(j, j^*)$ 的范围变窄，在胜出神经元附近的神经元数变少。因此，接着继续进行局部微调，空间分辨率提高。

自组织化映射有以下优点：

由于不需要映射内的相互结合，计算量少；

算法中不需要微分计算，数学上非常简单；

与生理学知识相适合。

图2.15是自组织化映射的生成例子。在这个例子中，输入向量

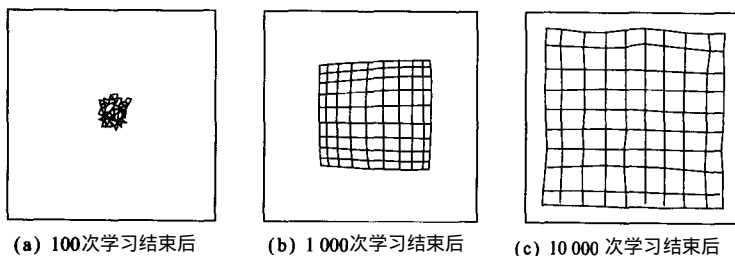


图 2.15 Kohonen 的自组织化映射示例

是 2 维，各自为从 0.0 到 1.0 的均匀随机数。图中，对映射层神经元的权值向量画线，邻接神经元用直线连接。这时，由于输入向量为均匀分布，完成的映射也将成为规则的正方格子形状。在有偏态分布的情况下，将形成与之对应的映射。

2.5 学习向量量化

2.5.1 学习向量量化

在进行自组织化映射时，在映射层配置比较多的神经元，把输入数据向二维层映射。在学习向量量化(LVQ)^[28,29]时，进行输入数据的模式分类。向量量化的想法是将数个数据归到一起，进行模式分类，以便完成信息压缩。

LVQ 可以认为是把自组织化映射改良成有教师学习的算法，它有以下特征：

- 学习算法简单；
- 理论上具有良好特性。

2.5.2 LVQ1 算法

LVQ 与自组织化映射相同，为双层网络结构。LVQ 网络以模式分类为目的，一般第 2 层比第 1 层的神经元数少。同一层的神经元之间没有结合。

在 LVQ 中第 2 层各个神经元有各自的领域。根据第 1 层和第 2 层之间的权值，输入向量被分到任意一个领域。Kohonen 把自组织化映射改良成有教师学习 首先设计了 LVQ1。下面说明 LVQ1 学习算法。

第 1 步：网络初始化：输入层和映射层之间的权值初始值，用较小的随机数设定。

第 2 步：输入向量的输入：首先，将输入向量

$$\mathbf{x} = [x_1, x_2, x_3, \dots, x_n]^T \quad (2.47)$$

输入给输入层。

第 3 步：在输出层计算与输入向量的距离：输出层中，计算各神经元的权值向量和各输入向量的距离。输出层第 j 个神经元和输入向量的距离，与自组织化映射的情况相同，由下式给出：

$$d_j = \sqrt{\sum_{i=1}^n (x_i - w_{ji})^2} \quad (2.48)$$

式中 w_{ji} 为输入层的 i 神经元和映射层的 j 神经元之间的权值。

第 4 步：与权值向量的距离为最小的神经元的选择： d_j 为最小，即选择使输入向量和权值向量的距离为最小的神经元。将其称为胜出神经元，记为 j^* 。

第 5 步：权值的学习：正确神经元成为胜出神经元时，即，正确分类时和不正确分类时的权值的更新公式是不同的，即

$$\Delta w_{ji} = \begin{cases} +\eta(x_i - w_{ji}) : \text{正确识别时} \\ -\eta(x_i - w_{ji}) : \text{错误识别时} \end{cases} \quad (2.49)$$

式中 $\eta > 0$ 为学习系数。

第 6 步 返回第 2 步 重复第 2 步到第 5 步，进行权值学习。

2.5.3 LVQ2 算法

当 LVQ1 出现识别错误时，即当不应该胜出的神经元以微差成为胜出神经元时，用 LVQ2 对其加以改良。这相当于输入向量恰好临近到两个神经元的领域的交界附近，以微差进入到错误一方。

具体算法中 第 5 步与 LVQ1 不同。

第 1 步～第 4 步：与 LVQ1 的场合相同。

第 5 步 权值的学习 (LVQ2)

网络在正确识别时 → 与 LVQ1 的情况相同，根据式 (2.49) 进行权值的更新。

当网络错误识别时 → 在满足以下条件时，对胜出神经元和第 2 位

神经元，根据式 (2.49) 进行权值的更新。

第 2 位神经元是正确的；

第 2 位神经元与胜出神经元的差很小。

作为开发者的 Kohonen 把 LVQ2 用于芬兰语声音识别，得到了比 LVQ1 优良的特性。而且，应用于声韵识别时，识别率可达到与 TDNN 同等的程度^[30]

第3章

互联型神经网络 的信号处理



- 3.1 互联型神经网络概要
- 3.2 霍普菲尔德网络
- 3.3 BAM及其改进
- 3.4 玻尔兹曼机和MFT学习

互联型神经网络，可以分为联想存储模型和用于模式识别及优化的网络。联想存储模型与计算机中使用的存储器有很大不同，它可以根据内容进行检索，类似于人的记忆方式，今后必然会得到广泛的应用。而后的网络特别适用于优化领域，并且已经获得了很多成果。本章，以互联型神经网络的基本原理为中心进行介绍。

3.1 互联型神经网络概要

联想存储模型包括 **Associatron**^[1]，福岛的时空间模式联想存储模型^[2] 霍普菲尔德网络 (Hopfield network)^[3] **HASP** (human associative processor)^[4] **BAM** (Bidirectional Associative Memory)^[5] 及 **MAM** (Multidirectional Associative Memory)^[6] 等形式。

用于模式识别或用于优化的神经网络中，玻尔兹曼机 (Boltzmann machine)^[7] 特别有名。

前章所述的层次型神经网络的信号流向为单向，而互联型神经网络为双向，这是一个很大的不同点 (参见注 3.1)。层次型神经网络主要用于模式识别，而互联型神经网络常被用于联想存储及最优化。

计算机通过设定有序的地址来存储记忆的内容。要将内容取出时，可以通过地址读取内容。对要存储的内容，进行符号化的变形，基本上以二进制信号的形式存储。

然而，人的大脑中没有地址，记忆的内容分散在神经回路网络上，被多重化存储。要取出时，根据线索回想。回想方法包括下述两种：

自回想线索和回想的内容相同时；

相互回想线索和回想的内容不同时。

如图 3.1 所示，自己回想是说，只看到面部的一部分就回想出全部面容的情况。相互回想比如说，在听到柠檬这个词时，想起了红茶。

www.aibbt.com 让未来触手可及

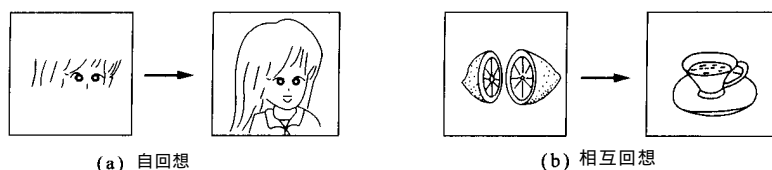


图 3.1 自回想与相互回想

注 3.1 神经网络有几个改进形及变形，因此，严密地讲，还不能断言信号的流向在层次型为单方向，相互结合型为双方向。

3.2 霍普菲尔德网络

霍普菲尔德网络，是美国的物理学家霍普菲尔德(J. J. Hopfield)在 1982 年提出的神经网络^[3]。他引入物理学能量函数的思想，对稳定性问题作出了一个解决方案，而且，应用这个网络还可使解法困难的组合最优化问题容易求解。

这里，在说明了应用相关的联想存储原理之后，介绍离散霍普菲尔德网络的工作原理。

3.2.1 基于相关的联想存储原理

首先说明基于相关矩阵的联想存储方式。由 N 个神经元互联的网络存储 P 个模式向量对 $(\mathbf{A}_i, \mathbf{B}_i)$ ($\mathbf{A}_i = [a_1^{(i)}, a_2^{(i)}, \dots, a_N^{(i)}]^T$, $\mathbf{B}_i = [b_1^{(i)}, b_2^{(i)}, \dots, b_N^{(i)}]^T$, $i=1, 2, \dots, P$)。即由 \mathbf{A}_i 回想 \mathbf{B}_i 的相互回想。神经元之间的权值用下列矩阵表示：

$$\begin{aligned}
 \mathbf{W} &= \sum_{i=1}^P \mathbf{B}_i \mathbf{A}_i^T \\
 &= \sum_{i=1}^P \begin{bmatrix} b_1^{(i)} \\ b_2^{(i)} \\ \vdots \\ b_N^{(i)} \end{bmatrix} [a_1^{(i)}, a_2^{(i)}, \dots, a_N^{(i)}]
 \end{aligned}$$

$$= \begin{bmatrix} \sum_{i=1}^P b_1^{(i)} a_1^{(i)} & \cdots & \sum_{i=1}^P b_1^{(i)} a_N^{(i)} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^P b_N^{(i)} a_1^{(i)} & \cdots & \sum_{i=1}^P b_N^{(i)} a_N^{(i)} \end{bmatrix} \quad (3.1)$$

式中 \mathbf{A}_i^T 表示 \mathbf{A}_i 的转置向量, 矩阵 \mathbf{W} 的 (i, j) 表示神经元 i 和神经元 j 之间的权值。

下面考虑回想问题。 \mathbf{A}_p 为输入 则输出为

$$\mathbf{W}\mathbf{A}_p = \sum_{i \neq p} \mathbf{B}_i \mathbf{A}_i^T \mathbf{A}_p + \mathbf{B}_p \mathbf{A}_p^T \mathbf{A}_p \quad (3.2)$$

这里假定模式间的相关性很低, 则右边第 1 项约为 0。第 2 项, 由于向量要素为 $\{-1, 1\}$, 所以由 $\mathbf{A}_p^T \mathbf{A}_p = N$, 有

$$\mathbf{W}\mathbf{A}_p \approx N\mathbf{B}_p \quad (3.3)$$

可见, 能够回想与 \mathbf{A}_p 对应的 \mathbf{B}_p 。

另外, 权值矩阵也可以根据下面的 Hebb 学习来求得:

$$\mathbf{W}^{\text{new}} = \mathbf{W}^{\text{old}} + \eta \mathbf{B}_i \mathbf{A}_i^T \quad (3.4)$$

直接用式(3.4)存在这样一个问题, 如果是连续学习权值矩阵将无限膨大。因此, 有对权值矩阵应用忘却系数 $\lambda (0 < \lambda < 1)$ 。

$$\mathbf{W}^{\text{new}} = \lambda \mathbf{W}^{\text{old}} + \eta \mathbf{B}_i \mathbf{A}_i^T \quad (3.5)$$

来防止权值矩阵发散。此外还有, 把 $\bar{\mathbf{A}}_i, \bar{\mathbf{B}}_i$ 分别作为为各自 $\mathbf{A}_i, \mathbf{B}_i$ 的平均向量, 有

$$\mathbf{W}^{\text{new}} = \mathbf{W}^{\text{old}} + \eta (\mathbf{B}_i - \bar{\mathbf{B}}_i)(\mathbf{A}_i - \bar{\mathbf{A}}_i)^T \quad (3.6)$$

用输入数据的分散成分进行学习的协方差(covariance)学习规则等。

3.2.2 霍普菲尔德网络

通过霍普菲尔德网络可进行自回想。霍普菲尔德网络的构成, 如图 3.2 所示, 各神经元为被结合的互联型。但是, 由于没有返回到输出的神经元本身的结合, 权值矩阵的对角成分为 0 ($w_{ii} = 0$)。

让 N 个神经元相互结合的霍普菲尔网络记忆 P 个模式向量 $\mathbf{A}_i = [a_1^{(i)}, a_2^{(i)}, \dots, a_N^{(i)}]^T (i = 1, 2, \dots, P)$ 。 N 个神经元之间的权值用下列矩阵表示：

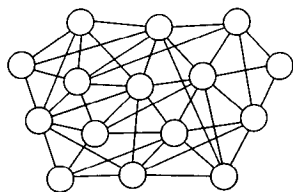


图 3.2 霍普菲尔网络的结构

$$\begin{aligned}
 \mathbf{W} &= \begin{bmatrix} 0 & w_{12} & \cdots & w_{1N} \\ w_{21} & \ddots & & \vdots \\ \vdots & & 0 & w_{N-1,N} \\ w_{N1} & \cdots & w_{N,N-1} & 0 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & \sum_{i=1}^P a_1^{(i)} a_2^{(i)} & \cdots & \sum_{i=1}^P a_1^{(i)} a_N^{(i)} \\ \sum_{i=1}^P a_2^{(i)} a_1^{(i)} & \ddots & & \vdots \\ \vdots & & 0 & \sum_{i=1}^P a_{N-1}^{(i)} a_N^{(i)} \\ \sum_{i=1}^P a_N^{(i)} a_1^{(i)} & \cdots & \sum_{i=1}^P a_N^{(i)} a_{N-1}^{(i)} & 0 \end{bmatrix}
 \end{aligned} \tag{3.7}$$

式中, $w_{ij} = w_{ji}$ 向量要素为 $\{0, 1\}$ 。各神经元根据下式进行状态转移：

$$y_i^{\text{new}} = \begin{cases} 1, & \sum_{j=1}^N w_{ij} y_j^{\text{old}} - \theta_i > 0 \\ y_i^{\text{old}}, & \sum_{j=1}^N w_{ij} y_j^{\text{old}} - \theta_i = 0 \\ 0, & \sum_{j=1}^N w_{ij} y_j^{\text{old}} - \theta_i < 0 \end{cases} \tag{3.8}$$

式中, $y_i (i=1, 2, \dots, N)$ 表示各神经元输出, θ_i 表示神经元 i 的阈值。

霍普菲尔网络, 根据与物理学的类推, 定义了下面的能量函数：

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} y_i y_j + \sum_{i=1}^N \theta_i y_i \quad (3.9)$$

当网络进行状态转移时，能量单调减少。因此，能量函数不会增加

为了进行表示，我们取当任意固定 i 时， y_i 的值从 1 到 0 变化。这样，能量 E 的变化量由式 (3.9) 可得

$$\begin{aligned} \Delta E_i &= E_{y_i=0} - E_{y_i=1} \\ &= \sum_{j=1}^N w_{ij} y_j - \theta_i \end{aligned} \quad (3.10)$$

式中，由于要使 y_i 的值为 0 则下式成立。

$$\sum_{j=1}^N w_{ij} y_j - \theta_i < 0 \quad (3.11)$$

因此 $E < 0$ 能量的确在减少。在 y_i 的变化相反的情况下，与从 0 到 1 变化的情况相同，能量也是在减少的。从图 3.3 可以看出，霍普菲尔德网络在取能量函数的极小值时为稳定状态。

霍普菲尔德网络用于联想存储时，已经证明其存储容量约为 $0.14N$ (N 为神经元数)^[8]，和其它学习方式的存储容量相比是比较低的。但是，在其余相关的学习中，由于不必多次重复学习同一数据，学习非常快。还有 Hebb 学习及突触的长期增强等与生理学知识相适合的部分还是很多的。

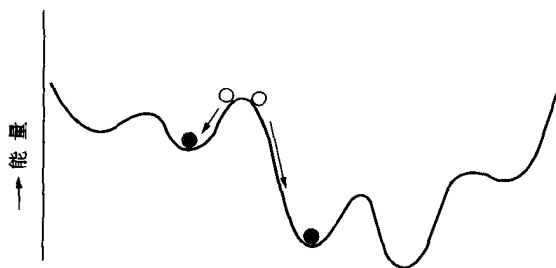


图 3.3 霍普菲尔德网络的平衡状态

3.3 BAM 及其改进

霍普菲尔德网络是自己回想形式的联想存储模型。这里要介绍的双向联想存储 (BAM) 模型, 是相互回想形式的联想存储模型。本节首先介绍 BAM 基本形式的构成和工作原理^[5] 然后 对可以增加存储容量的 PRLAB(pseudo-relaxation learning algorithm for BAM) 即称为 BAM 用仿射松弛学习算法^[9] 予以说明。此外, 还介绍 PRLAB 和相关学习相组合的缩短学习时间的快速学习算法 (quick learning algorithm for BAM)^[10]。

3.3.1 BAM 的构成及工作原理

图 3.4 表示的是 BAM 的网络构成。BAM 由两层组成, 层之间全部相结合, 而层内没有结合。

首先说明 BAM 的记忆过程。

考虑第 1 层 N 个神经元和第 2 层 M 个神经元构成的 NM -BAM。让这个网络记忆 P 个模式向量对 (A_i, B_i) ($A_i = [a_1^{(i)}, a_2^{(i)}, \dots, a_N^{(i)}]^T, B_i = [b_1^{(i)}, b_2^{(i)}, \dots, b_M^{(i)}]^T, i=1, 2, \dots, P$)。与 3.2.1 节的情况相同, 可以根据相关矩阵来存储。对于 $A_i \rightarrow B_i$ 的方向存储, 用下面的相关矩阵:

$$W = \sum_{i=1}^P A_i B_i^T \quad (3.12)$$

对于反方向, 即 $B_i \rightarrow A_i$ 相关矩阵为下面的转置矩阵:

$$W^T = \sum_{i=1}^P B_i A_i^T \quad (3.13)$$

接下来说明回想过程。在回想时, 可根据某个输入, 使得第 1 层输出为 $x = [x_1, x_2, \dots, x_N]^T$ 。在第 2 层的各神经元中,

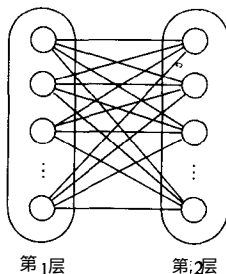


图 3.4 BAM 网络结构

先求输入和。对神经元 j 的输入和按下式给出：

$$S_{yj} = \sum_{i=1}^N w_{ji} x_i - \theta_{yj} \quad (3.14)$$

式中 θ_{yj} 为第 2 层神经元 j 的阈值。

第 2 层输出神经元 j 的输出根据下式确定：

$$y_j^{\text{new}} = \begin{cases} 1, & S_{yj} > 0 \\ y_j^{\text{old}}, & S_{yj} = 0 \\ -1, & S_{yj} < 0 \end{cases} \quad (3.15)$$

第 2 层的输出传向第 1 层，用同样的方法可确定第 1 层的输出。让这样的动作，即回响 (reverberation) 重复进行，网络就能达到平衡状态。

注 3.2 与霍普菲尔德网络的情况相同，BAM 的情况也可以定义能量函数，向极小点收敛^[5]。BAM 与 3.2.1 中说明的相互回想模型在网络结构上不同，但是，原理上极其相近。因此，存储容量也很低。在下一节中，介绍能够较大增加存储容量的学习算法。

3.3.2 PRLAB

PRLAB 的原理是基于联立不等式的解法，而在 BAM 中由式 (3.15) 可知主要注重输入和的符号。

在这里首先介绍解联立不等式的松弛法，然后是如何根据用仿射松弛法来大幅度提高收敛的速度，最后，说明把仿射松弛法应用于 BAM 的学习算法 (快速学习) 及其特性。

(1) 联立不等式的松弛法的解法考察下面的 m 元联立不等式：

$$w^T x^i + b^i \geq 0, \quad i=1, 2, \dots, m \quad (3.16)$$

式中, $w \in R^n, x^i \in R^n, b^i \in R$ 。

对于各个不等式，是以分成两部分的空间 R^n 的一方作为值域。

$$H^i = \{x^i \mid w^T x^i + b^i \geq 0\}, \quad i=1, 2, \dots, m \quad (3.17)$$

H^i 的共同值域为值域 C 。因此

$$C = \bigcap_{i=1}^m H^i \quad (3.18)$$

下面，说明根据松弛法求解这样的联立不等式的方法。为简便起见，以二元联立不等式为例。如图 3.5 所示 对于 w 向量 如果重复向 H^1 和 H^2 作垂线，就逐渐逼近 H^1 和 H^2 的交点。将其用公式表示为

$$w^{\text{new}} = w^{\text{old}} - \frac{(w^{\text{old}})^T x^i + b^i}{\|x^i\|} \frac{x^i}{\|x^i\|} \quad (3.19)$$

上式右边第 2 项的开始部分表示 w^{old} 到 H^1 的距离，后面部分表示修正方向的单位向量。这里，“ $\|\cdot\|$ ”表示向量大小的范数。

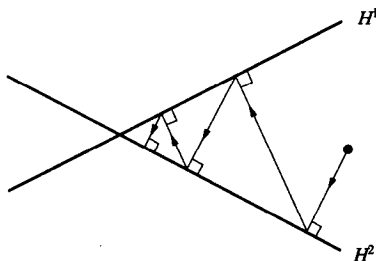


图 3.5 使用松弛法的不等式解法 (收敛快的例子)

实际上，为了收敛的稳定化，引入松弛常数 $\lambda (0 < \lambda < 2)$ ：

$$w^{\text{new}} = w^{\text{old}} - \lambda \frac{(w^{\text{old}})^T x^i + b^i}{\|x^i\|^2} x^i \quad (3.20)$$

但对于图 3.6 所示的情况，由于二个平面形成的角度较小，即使重复这样的操作，也很难向交点逼近。可见在 H^1 和 H^2 的角度较小的情况下，收敛极其缓慢。

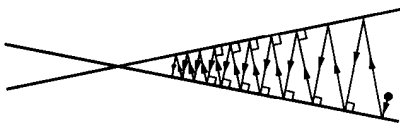


图 3.6 使用松弛法的不等式解法 (收敛慢的例子)

(2) 仿射松弛法的导入为了解决上述问题，在仿射松弛法中，如图 3.7 所示先将 H^i 的位置向值域的一侧移动 $\xi(>0)$ 然后应用松弛法。

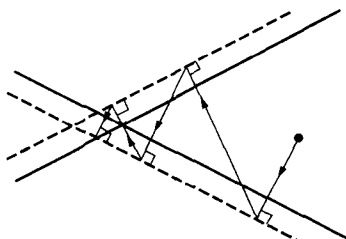


图 3.7 使用仿射松弛法的不等式解法

应用式(3.16),可得

$$w^T x^i + b^i - \xi \geq 0, i = 1, 2, \dots, m \quad (3.21)$$

这样做，不仅可以超过交点进入解域，而且可以快速收敛。对式(3.21)应用仿射松弛法可得

$$w^{new} = w^{old} - \lambda \frac{(w^{old})^T x^i + b^i - \xi}{\|x^i\|^2} x^i \quad (3.22)$$

根据式(3.21),式(3.22),不满足不等式而进行学习时,即当

$$(w^{old})^T x^i + b^i < 0 \quad (3.23)$$

时,为

$$|(w^{old})^T x^i + b^i - \xi| > |(w^{old})^T x^i + b^i| \quad (3.24)$$

因此,可见看出,应用仿射松弛法时,与被计算的修正量相比,绝对值增大了 ξ 。

(3) 将仿射松弛法引入 BAM 首先讨论松弛法的 BMA 学习算法的公式。

如上所述,BAM 存储的是模式对。当给出某个模式对时,考虑到第 2 层的神经元,因此来自第 1 层的输入和的符号应当和给予其神经元的要素的符号相一致。对于全部模式对,必须使

$$\left(\sum_{i=1}^N w_{ji} x_i - \theta_{yj} \right) y_j > 0, \quad j = 1, 2, \dots, M \quad (3.25)$$

成立。同样,对于第 1 层神经元 必须使

$$\left(\sum_{j=1}^M w_{ij} y_j - \theta_{xi}\right) x_i > 0, \quad i = 1, 2, \dots, N \quad (3.26)$$

成立。式中, θ_{xi}, θ_{yj} 分别为第 1 层神经元 i 和第 2 层神经元 j 的阈值。

如果式 (3.25) 和式 (3.26) 成立, 就可以保证被存储的全部模式对的回想。因此, 只限于在这些不等式不成立的情况下使用仿射松弛法。也就是说, 仅当某个神经元的输入和与其神经元应该输出的值, 符号不同时, 进行权值更新。

还有 为了得到式 (3.24) 的效果, 权值更新量必须要比不用仿射松弛法时大。

考虑以上几点并结合式 (3.22) 相同的情况, 得到用于 BAM 的仿射松弛学习算法。

首先 从第 2 层神经元考虑。对第 2 层神经元 j 的输入和 用

$$S_{yj} = \sum_{i=1}^N w_{ji} x_i - \theta_{yj} \quad (3.27)$$

给出。权值更新, 限于 $S_{yj} y_j \leq 0$, 根据下式进行。这里, $x = [x_1, x_2, \dots, x_N]^T$ 。

$$w_{ij}^{\text{new}} = w_{ij}^{\text{old}} - \lambda \frac{S_{yj} - \xi y_j}{\|x\|^2} x_i \quad (3.28)$$

由式 (3.28) 可见, $S_{yj} y_j < 0$ 时, 由于 S_{yj} 的符号和 y_j 的符号相反, 有

$$|S_{yj} - \xi y_j| > |S_{yj}| \quad (3.29)$$

阈值的更新也在 $S_{yj} y_j \leq 0$ 的情况下进行。

$$\theta_{yj}^{\text{new}} = \theta_{yj}^{\text{old}} + \lambda \frac{S_{yj} - \xi y_j}{\|x\|^2} \quad (3.30)$$

上式右边第 2 项的符号为 (+) 是因为 用式 (3.14) 的神经元的输入和定义 阈值符号取 (-) 的缘故。

同样, 对于第 1 层神经元的学习算法也按以下来求。这里, $y = [y_1, y_2, \dots, y_M]^T$ 。

$$w_{ij}^{\text{new}} = w_{ij}^{\text{old}} - \lambda \frac{S_{xi} - \xi x_i}{\|y\|^2} y_j, \quad S_{xi} x_i \leq 0 \text{ 时} \quad (3.31)$$

$$\theta_{xi}^{new} = \theta_{xi}^{old} + \lambda \frac{S_{xi} - \xi x_i}{\|y\|^2}, S_{xi} x_i \leq 0 \text{ 时} \quad (3.32)$$

其中

$$S_{xi} = \sum_{j=1}^M w_{ij} y_j - \theta_{xi} \quad (3.33)$$

(4) 快速学习在 PRLAB 中，原理上是通过求解联立不等式来求权值的。这时，权值的初始值取较小的随机数。如果用 PRLAB 求解的权值和用相关学习求解的权值在一定程度上接近的话，是否可以把用相关学习求得的权值作为 PRLAB 的初始值呢，快速学习就是基于这样的想法。

快速学习为下面的两段学习方式。

确定权值初始值：由相关学习求权值初始值；

权值的学习：根据 PRLAB 算法进行权值学习。

由于相关学习进行一次就结束，计算量非常小。很多的仿真结果证实，快速学习，在缩短学习时间，抗噪声等方面比较有效^[1-3]。

图 3.8 表示在 50-50BAM 情况下的存储模式的相关和学习次数的关系。存储模式的相关越高，学习时间越长，但通过快速学习却可以对其增加进行抑制。

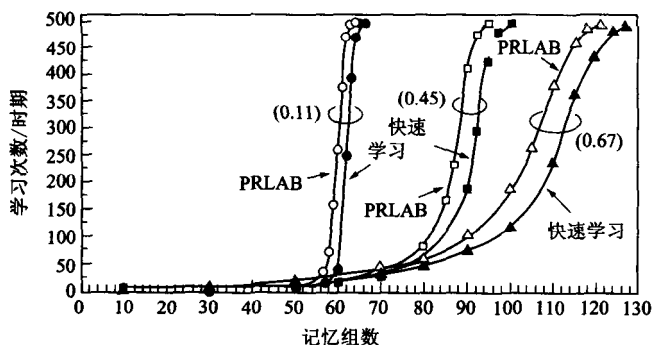


图 3.8 PRLAB 与快速学习的学习时间比较

过去的联想存储模型，存储容量太小，限制了其进行实用化。通过快速学习极大缓和了这种限制，可以预见联想存储模型今后将会得到广泛应用。

3.4 玻尔兹曼机和MFT学习

过去的神经网络的任何一个动作都是确定的，不需要进行概率方面的考虑。比如在霍普菲尔德网络中，由于能量是单调减少的，结果必将会落入初始值附近的极小值。但是，却不能保证这就是最小值。

在玻尔兹曼机^[7]中，应用退火的思想可以高概率来求最小解。但是，这需要很长的计算时间。因此，应用物理学中的平均场理论(MFT; Mean Field Theory)将概率性的玻尔兹曼机的动作近似成确定性的动作，提出了可以大幅度减少必要计算量的 **MFT** 学习算法。这里，首先解说玻尔兹曼机的学习算法，然后说明 MFT 学习。

3.4.1 玻尔兹曼机的学习算法概要

下面举一个例子来说明玻尔兹曼机学习算法的退火的思想。

有一个凹凸不平的盆，要使一个小球稳定在最低的地方。如果把小球轻轻地放入盆中，那么结果必然是小球稳定在距放入地方最近的低洼处。

但是，稳定所在的地方并不保证是最低的地方。这一动作与霍普菲尔德网络相当。在玻尔兹曼机中，大幅度摇晃刚放入小球的盆子，然后逐渐减小摇晃的幅度。这样，小球才有可能到达盆子最低处。

3.4.2 玻尔兹曼分布

在玻尔兹曼机中，将上述动作通过称为温度 T 的参数来进行。现在只考虑一个神经元状态的两个不同状态 α, β 。各自状态的网络能量分别为 E_α 和 E_β 。网络从状态 α 向状态 β 变化的概率，根据它们的能

量差和温度 T , 由下式确定:

$$p_{\beta \rightarrow \alpha} = \frac{1}{1 + \exp\left(-\frac{E_{\alpha} - E_{\beta}}{T}\right)} \quad (3.34)$$

为简单起见, 式中省略了玻尔兹曼常数。

玻尔兹曼机中用的能量, 与霍普菲尔德网络的情况相同。因此, 任意神经元 y_i 的输出取从 $1 \sim 0$ 变化, 这时的能量差, 由式 (3.9) 式 (3.10) 变为

$$\begin{aligned} \Delta E_i &= E_{\alpha} - E_{\beta} \\ &= \sum_{j=1}^N w_{ij} y_j - \theta_i \end{aligned} \quad (3.35)$$

这时, 神经元 y_i 的输出值为 1 的概率为

$$p_{y_i=1} = \frac{1}{1 + \exp\left(-\frac{\Delta E_i}{T}\right)} \quad (3.36)$$

还有 神经元 y_i 的输出值为 0 的概率为

$$\begin{aligned} p_{y_i=0} &= 1 - p_{y_i=1} \\ &= \frac{1}{1 + \exp\left(\frac{\Delta E_i}{T}\right)} \end{aligned} \quad (3.37)$$

图 3.9 中表示式 (3.36) 的大概形状。可以看出, 温度 T 比较高

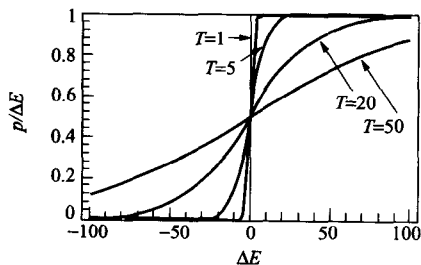


图 3.9 $p(x) = 1 / \left[1 + \exp\left(-\frac{x}{T}\right) \right]$ 大概形状

状态变化接近随机，随着温度变低向确定性的动作靠近。

在网络中被考虑的全部状态中，成为状态 α, β 的个数分别为 N_α, N_β 。在平衡状态中，从状态 α 向状态 β 的变化量及其相反方向，即从状态 β 向状态 α 的变化量相等。因此，有

$$p_{\beta \leftarrow \alpha} \cdot N_\alpha = p_{\alpha \leftarrow \beta} \cdot N_\beta \quad (3.38)$$

由式(3.34)和式(3.38)，下式成立。

$$\begin{aligned} \frac{N_\alpha}{N_\beta} &= \frac{p_{\alpha \leftarrow \beta}}{p_{\beta \leftarrow \alpha}} = \exp\left(-\frac{E_\alpha - E_\beta}{T}\right) \\ &= \exp\left(-\frac{E_\alpha}{T}\right) \exp\left(\frac{E_\beta}{T}\right) \end{aligned} \quad (3.39)$$

因此，在网络中考虑到全部状态 $\alpha, \beta, \gamma, \dots$ 取状态 α 的概率为

$$\begin{aligned} p_\alpha &= \frac{N_\alpha}{N_\alpha + N_\beta + N_\gamma + \dots} \\ &= \frac{\exp\left(-\frac{E_\alpha}{T}\right)}{Z} \end{aligned} \quad (3.40)$$

式中

$$Z = \sum_{\lambda} \exp\left(-\frac{E_\lambda}{T}\right) \quad (3.41)$$

式(3.40)和式(3.41)是统计力学中已知的玻尔兹曼分布(Boltzmann distribution)，玻尔兹曼机由此而得名。由式(3.40)可见，能量状态低，出现的概率比较大。

3.4.3 玻尔兹曼机的学习算法

玻尔兹曼机是为了使网络实际输出的分布接近于理想输出的概率分布，而进行权值的学习。

把输入神经元和输出神经元称为可视神经元(visible neuron)其它神经元称为隐含神经元(hidden neuron)。没有外部来的输入时，可视神经元群的状态 α 的概率分布为 P_α ，给出从外部来的输入输出时，可视神经元群的状态 α 的概率分布为 R_α 。

通过权值的学习，达到使这二个概率分布的区别尽可能的小。作为表示概率分布之间距离的尺度，有一个称为 **Kullback** 的信息量 (Kullback's information) G 的量。在玻尔兹曼机中为下述形式：

$$G = \sum_a R_a \log \frac{R_a}{P_a} \quad (3.42)$$

由上式可以看出，在 P_a 和 R_a 的概率分布完全相等的情况下， G 最小，为 0。

玻尔兹曼机的学习规则，根据最速下降法，把 Kullback 的信息量 G 对权值 w_{ij} 偏微分来求得

$$\Delta w_{ij} = -\eta \frac{\partial G}{\partial w_{ij}} \quad (3.43)$$

若只表示结果，就为

$$\Delta w_{ij} = \frac{\eta}{T} [\overline{\langle y_i y_j \rangle}_{\text{clamped}} - \overline{\langle y_i y_j \rangle}_{\text{free}}] \quad (3.44)$$

式中， $\overline{\langle y_i y_j \rangle}_{\text{clamped}}$ 表示把可视神经元固定 (clamp) 时神经元 i 和神经元 j 同时为 1 的概率， $\overline{\langle y_i y_j \rangle}_{\text{free}}$ 表示把可视神经元开放 (free) 时神经元 i 和神经元 j 同时为 1 的概率。

下面，将玻尔兹曼机的实际学习算法进行一下概括：

第 1 步 网络的初始化 权值的初始值用较小的随机数设定 然后确定退火计划，即温度 T 的降低方法。

第 2 步：固定阶段（可视神经元的固定）与训练数据相适应 固定可视神经元。

随机选择隐含神经元，根据式 (3.35)～式 (3.37) 改变神经元的状态。在还未达到各温度的平衡状态时多次进行，与此同时根据退火计划降低温度 T 。

到达最终温度 T_0 后，求神经元 i 和神经元 j 同时为 1 的概率 $\overline{\langle y_i y_j \rangle}_{\text{clamped}}$ 。通过全部神经元的组合来进行。

第 3 步 开放阶段（可视神经元的开放）开放可视神经元 可以取

可视神经元的 0 和 1 中的任意一个值。

与第 2 步的情况相同，隐神经元随机选择，根据式 (3.35)~式 (3.37) 改变神经元的状态。在未达到各温度的平衡状态时多次进行，与此同时根据退火计划降低温度 T 。

到达最终温度 T_0 后，求神经元 i 和神经元 j 同时为 1 的概率 $\langle y_i y_j \rangle_{\text{free}}$ 。通过全部神经元的组合来进行。

第 4 步 权值的学习 根据式 (3.44) 进行权值的更新。

第 5 步 重复 查看是否满足事先确定的收敛条件。若不满足 返回第 2 步 重复学习。

由于玻尔兹曼机的基础是统计力学，所以理论明晰。但是，使用时，由于必须在达到各温度平衡状态前进行状态变化，存在计算时间非常长的问题。因此，应用的实例不太多。

然而，对于优化领域，由于不需要学习而只求平衡状态的情况很多 在这样的简化问题中 经常应用玻尔兹曼机^[11]

3.4.4 MFT 学习算法

MFT 是物理学中用到的近似理论，图 3.10 表示了 MFT 的基本

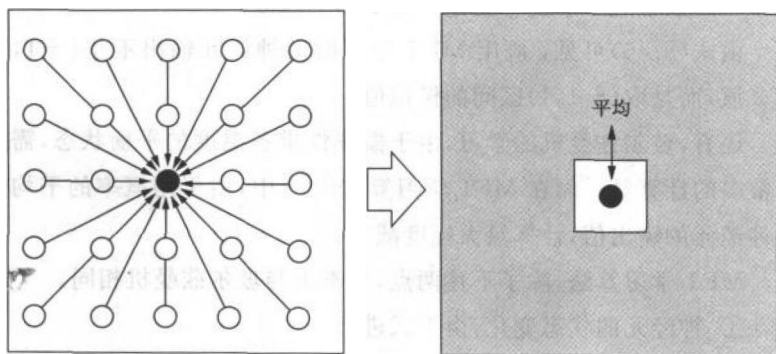


图 3.10 MFT 的基本思想

思想。其方法是基于“考虑来自周围物质的影响时，不是分别考虑来自各自的影响，而是以全部的平均影响度近似”这样的想法。

在前面已经说明的玻尔兹曼机中，各神经元的输出为 $\{0,1\}$ 的二值，输出概率分别按式 (3.36) 式 (3.37) 确定。而在基本 MFT 学习中神经元输出为 $\{-1,1\}$ 的二值，若用各自的输出概率来求神经元输出的期望值 $\langle y_i \rangle$ 就为

$$\begin{aligned}
 \langle y_i \rangle &= p_{y_i=+1} \cdot (+1) + p_{y_i=-1} \cdot (-1) \\
 &= \frac{1}{1 + \exp\left(-\frac{\Delta E_i}{T}\right)} - \frac{1}{1 + \exp\left(\frac{\Delta E_i}{T}\right)} \\
 &= \frac{\exp\left(\frac{\Delta E_i}{2T}\right) - \exp\left(-\frac{\Delta E_i}{2T}\right)}{\exp\left(\frac{\Delta E_i}{2T}\right) + \exp\left(-\frac{\Delta E_i}{2T}\right)} \\
 &= \tanh\left(\frac{\Delta E_i}{2T}\right) \\
 &= \tanh\left[\frac{\sum_j w_{ij} y_j}{T}\right] \tag{3.45}
 \end{aligned}$$

最后的式子中， T 的系数 2 之所以消失，是因为神经元输出为 $\{-1,1\}$ ，数学式子中能量变化量为 2 倍的缘故。

由式 (3.45) 可见，应用 MFT 学习时的神经元输出不是 $\{-1,1\}$ 的二值而是取 $(-1,1)$ 区间的模拟值。

还有，玻尔兹曼机的学习，由于概率性求各温度的平衡状态，需要非常多的计算量。而在 MFT 学习算法^[12,13] 中，由于用概率的平均值求神经元的输出值，计算量大幅度减少。

MFT 学习算法，除了下述两点，基本上与玻尔兹曼机相同。

神经元的状态变化，由下式进行：

$$y_i = \tanh\left[\frac{\sum_j w_{ij} y_j}{T}\right] \tag{3.46}$$

因此，不需要平均计算，权值的更新公式变为下式：

$$\Delta w_{ij} = \frac{\eta}{T} [(y_i y_j)_{\text{clamped}} - (y_i y_j)_{\text{free}}] \quad (3.47)$$

注 3.3 MFT 学习算法，用于模式识别的情况下，与反向传播和玻尔兹曼机的特性没有多大差别。但用于联想存储模型时，根据文献 [12,14] 可知，存储容量为隐含神经元数的 10 倍到 20 倍。

第4章

模糊性和模糊集合

- 4.1 模糊集合的思想
- 4.2 模糊集合与模糊性的表示
- 4.3 模糊集合的运算
- 4.4 概念模糊集合
- 4.5 模糊度和不确定性

通过应用模糊理论，就可以在数学上处理模糊性概念。模糊理论的基础是模糊集合，引入隶属函数的概念后，模糊性概念的数学表示及运算就成为可能。本章，将对用于表示模糊性的模糊集合的引入，模糊集合的基本运算以及概念模糊集合等进行介绍。

4.1 模糊集合的思想

模糊集合^[1]通过表示对集合的归属程度的 $[0,1]$ 这样的隶属值，尝试表示没有明确界限的模糊概念。这与通常集合的属于还是不属于的 Yes/No 判断基准的定义相对，由于引入归属的程度，模糊集合相当于允许不明确性。

在对数值信号进行评价，向人类可以直接理解的符号变换时，这个不明确性程度的隶属值就变得很重要。

下面，简要介绍智能信号处理中重要的模糊集合和模糊性的表示及其基本运算，还有与神经网络的信号处理的关联和向概念模糊集合的扩展等。

4.2 模糊集合与模糊性的表示

在我们日常进行的对话中，经常会有“热”、“冷”这样的本身含有主观模糊性的概念，此外，在平常的会话中也含有主观模糊性。札德在 1965 年提出了含有这样的主观意义的模糊知识的模糊理论^[1]，并且给出了其数学表示和基本运算方法，提出了推理的方法。

E. H. Mamdani 在 1974 年，为了把专家的控制知识用语言性的 if-then 规则进行表示，用推理来进行控制，而使用了札德倡导的模糊理论^[2]。这是模糊控制的开始。以后，模糊控制在理论和应用两个方

面进行了诸多研究，近年来已被应用到很多系统中。

4.2.1 隶属函数

模糊系统中，要表示像“热”这样的主观意思时，模糊集合为 A ，隶属函数定义如下。首先，若隶属函数为 $\mu_A(x)$ ，气温为 x 时，隶属函数如图 4.1 所示。

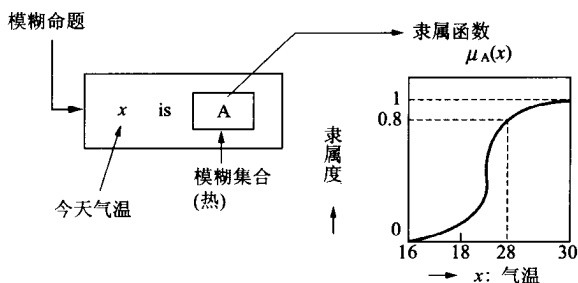


图 4.1 模糊集合与隶属函数

“今天很热”这样的命题中，今天的气温为 28 的情况下，其隶属等级表示为 0.8。而且，在前述的模糊规则中也存在的，在模糊控制中经常用到的模糊集合有：

PB (Positive Big)(正大)

PM (Positive Medium)(正中)

PS (Positive Small)(正小)

ZO (Zero)(零)

NS (Negative Small)(负小)

NM (Negative Medium)(负中)

NB (Negative Big)(负大)

这些隶属函数包括如图 4.2 这样的用一元式合成的二角形，用二元式合成的倒钟形，还有聚集多点的离散型等。

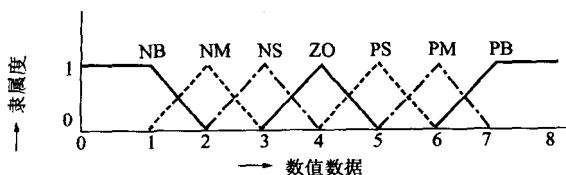


图 4.2 隶属函数的一个例子

关于这些区分数、范围、形状等，专家根据自己的主观选择使用，用隶属度函数来表示所要表现的模糊集和所具有的意义。

评价这些被输入数值信号的隶属函数，要用到单调增函数及单调减函数，或者由它们的组合形成的凸函数。根据隶属函数，把被输入的数值信号变换成相应的“符号”和“模糊性程度”。

4.2.2 n -cube 的模糊性表示

B. Kosko 根据与神经网络的关联，用图 4.3 所示的 n -cube 来表示模糊性。该图是天气预报中关于“热”和“降雨”的二维例子在平面上表示。事件为 n 维的情况下用 n -cube。图中， x 轴表示“热”的可信度， y 轴表示“降雨”的可信度。点 O 是最模糊点。点 P 表示某个天气预报的结果，即“预报为晴天，变热”。在模糊集和模糊逻辑及基于模糊推理的信号处理中，可以把这样的“符号”标识和表示其“模糊性”。

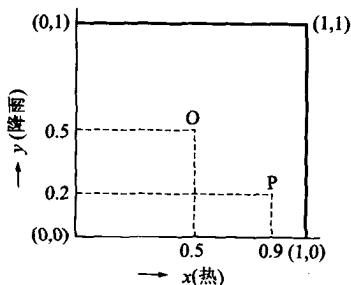


图 4.3 n -cube 的模糊性表示 (天气预报的例子)

程度'的可信度的由 0 到 1 围成的立方体向别的立方体映射。

4.2.3 模糊系统和神经网络的共同点

对于把神经元的激活值限定在 $[0,1]$ 区间的情况下,由于可以和附加给模糊集合各要素的隶属值同等看待,所以可以比较简单地把两个过程合成,或把任一过程的一部分用另一过程替换。

根据 Kosko 所述,由 n 个神经元组成的场信号处理状态空间,可以用 n 维超立方体表示,立方体内的 1 个点与把 n 个要素作为全体集合的模糊集之一相对应^[3]。即神经网络等同于模糊系统都是在同一单位空间内进行处理。

4.3 模糊集合的运算

与把'符号'标识用真/假(即 1/0)进行逻辑运算的二值逻辑相对应,在模糊集的逻辑演算中,必须考虑从 0 到 1 表示'模糊程度'的可信度。基本运算中,如图 4.4 所表示的,有模糊补集、模糊并集和模糊交集。本节运用 G. J. Klir 提出的公理,对模糊信号处理中必要的基本运算予以简单说明。

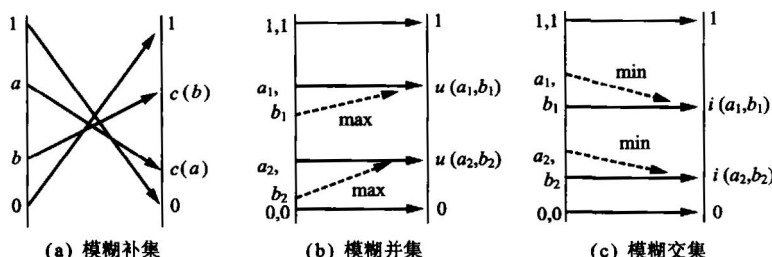


图 4.4 模糊集合运算概要

4.3.1 模糊补集

模糊补集的演算用下面的函数 c 定义：

$$c: [0, 1] \rightarrow [0, 1] \quad (4.1)$$

这里，函数 c 对于模糊集的隶属函数 $\mu_A(x)$ 用满足以下公理的补集运算 $c(\mu_A(x))$ 定义：

公理 1: $c(0)=1, c(1)=0$ (边界条件)

公理 2 如果 $a < b, a, b \in [0, 1]$ 则 $c(a) \geq c(b)$ (单调性)

公理 1 为与没有模糊性的普通集合对应的边界条件，也是普通集合要满足的条件。公理 2 在有顺序关系条件下，函数 c 成为单调非增。满足这两个公理的函数有很多。以下的运算经常用到：

$$\mu_c(x) = 1 - \mu_A(x) \quad (4.2)$$

4.3.2 模糊并集

模糊并集的演算用下面的函数 u 定义：

$$u: [0, 1] \times [0, 1] \rightarrow [0, 1] \quad (4.3)$$

这里，函数 u 的对应于模糊集 A, B 的隶属函数 $\mu_A(x), \mu_B(x)$ 用满足下面公理 3~6 并集的演算 $u(\mu_A(x), \mu_B(x))$ 定义。

公理 3: $u(0, 0)=0, u(0, 1)=u(1, 0)=u(1, 1)=1$ (边界条件)

一般化为, $u(a, 0)=u(0, a)=a, a \in [0, 1]$

公理 4: $u(a, b)=u(b, a)$ (可交换性)

公理 5: $a_1 \leq a_2$ 并且 $b_1 \leq b_2$, 如果 $a_1, a_2, b_1, b_2 \in [0, 1]$ 则

$u(a_1, b_1) \leq u(a_2, b_2)$ (单调性)

公理 6: $u(u(a, b), c)=u(a, u(b, c)), a, b, c \in [0, 1]$ (结合性)

公理 3 为与没有模糊性的普通集合对应的边界条件，也是普通集合要满足的条件。满足公理 3~6 的函数很多。如 T -范数运算符^[5]等(参见 5.2 节)。而且，经常要用到的是用下面的 ∇ 表示的 \max 运算符。

$$\mu_U(x) = \max\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \nabla \mu_B(x) \quad (4.4)$$

4.3.3 模糊交集

模糊交集的运算用下面的函数 i 定义:

$$i: [0,1] \times [0,1] \rightarrow [0,1] \quad (4.5)$$

这里, 函数 i 为对应于模糊集 A, B 的隶属函数 $\mu_A(x), \mu_B(x)$, 用满足下面公理 7~10 并集演算 $i(\mu_A(x), \mu_B(x))$ 定义:

公理 7: $i(1,1)=1, i(0,1)=i(1,0)=i(0,0)=0$ (边界条件)

一般化为, $i(a,1)=i(1,a)=a, a \in [0,1]$

公理 8: $i(a,b)=i(b,a)$ (可交换性)

公理 9: $a_1 \leq a_2$ 并且 $b_1 \leq b_2$, 如果 $a_1, a_2, b_1, b_2 \in [0,1]$, 则

$i(a_1, b_1) \leq i(a_2, b_2)$ (单调性)

公理 10: $i(i(a,b),c)=i(a,i(b,c)), a,b,c \in [0,1]$ (结合性)

公理 7 为与没有模糊性的普通集合对应的边界条件, 也是普通集合要满足的条件。满足公理 7~10 的函数有很多。一般有 T -准则运算符。而且, 经常要用到的有用下面的 \min 运算符。

$$\mu_1(x) = \min\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \Delta \mu_B(x) \quad (4.6)$$

4.4 概念模糊集合

前面所述的模糊集和的标识附以概念这个词, 模糊集合的形式就表示了概念的意思。若按照 Wittgenstein 的意思使用学说, 语言的意思可以由其他的语言来表示, 但其意思根据使用的场合而变化^[6]。

因此 如图 4.5 所示, 如果把使用时的注视范围用附带活性值的语言来表示的话, 就可以用由其他语言变形的模糊集合把模糊语言的意思分散表示。

把这个分散的知识表示称为概念模糊集合(CFS)。激活值的分布, 由于随各种状况而变化, 因此就可以表示状况的意思的变化。

而且 由于 CFS 不只局限于逻辑表示还能进行知识表示, 所以

CFS 能够明确表示逻辑上不能明确表示的知识。CFS 的激活值根据神经网络的联想存储进行控制。

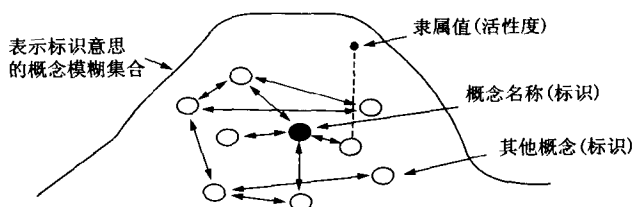


图 4.5 概念模糊集合的概要

因此，节点表示概念，连接则与概念之间关系的强度对应。若把应该被表示意思的概念节点激活，经过事先构造的网络，激活值就进行传播，引起反应动作。

这个反应动作在系统能量达到最小的地方停止，其结果，表示意思方面的概念节点，以被某等级激活的形式进行回想。这里被表示的激活值的分布为模糊集合，包含控制激活值机构的系统全体是 CFS。这里，根据使用时有限制的明确性，使用双向联想存储（BAM）参见 3.3 节）。BAM 中的激活值的传播及反应动作作用

$$\mathbf{A}(t+1) = \phi(\mathbf{M}^T \mathbf{B}(t)) \quad (4.7)$$

$$\mathbf{B}(t) = \phi(\mathbf{M} \mathbf{A}(t)) \quad (4.8)$$

来进行。

$\mathbf{M} \in R^{n \times p}$ 为联想矩阵 $\mathbf{A}(t) = [a_1(t), a_2(t), \dots, a_n(t)]^T$, $\mathbf{B}(t) = [b_1(t), b_2, \dots, b_n(t)]^T$ 为第 t 步中各节点的激活值向量，取值为 $[0, 1]$ 。 ϕ 为各节点的函数，取双弯曲函数。

使用神经网络的学习规则归纳性构造 CFS。也就是说，根据概念表示的事例，根据 Hebb 学习规则，修正节点之间的连接。本学习方式，如果考虑到神经网络的特征，也能够由含有噪声的示例进行学习。

CFS 对概念说明不需要的其他概念则不予以激活。因此，即使是从冗长的说明要素开始，就已经变成了包含节点选择的结构的学习。

还有，CFS 像我们通常形成概念时一样，分别把形成的一个个概念 M 依次组合的同时，形成全体的概念。即可以把由数个 $M_i (i=1, 2, \dots, n \text{ 个})$ 组成的 CFS 按下式合成：

$$M = \text{norm}[M_1 + M_2 + \dots + M_n] \quad (4.9)$$

式中， norm 为联想矩阵的正规化运算（参见第 5 章）。

在一般的神经网络中，学习结果中不进行明确的知识表示，而本方法则有获得明确表示的优点。

例 4.1 为了说明基于学习的 CFS 的归纳构成，这里举出包含状况依存性的 CFS 的例子。比如模糊集合“个子高”根据其对象是“日本人”还是“美国人”隶属函数的形状进行变化。因此这是依存于人的模糊集合。

这里用 Hebb 学习规则的公式归纳性构成联想矩阵，得到的个别 CFS 表示在图 4.6(a)、(b) 合成后得到的 CFS 用图 4.7 表示。

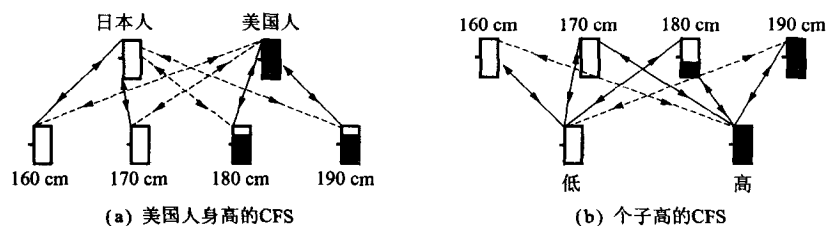


图 4.6 CFS 的例子

图 4.6 和图 4.7 中由关于人种的层（“日本人”；“美国人”）关于身高尺寸的层（160cm, 170cm, 180cm, 190cm）关于身高的层（高、低）这样的结构成了 CFS。用各个四角形中的黑面积表示激活值。

还有，根据其激活值动态表现的模糊集合。对于图 4.7 的各层之

间的联想矩阵，用 Hebb 学习规则归纳求出图 4.6 所示的个别联想矩阵，然后由合成来构成。

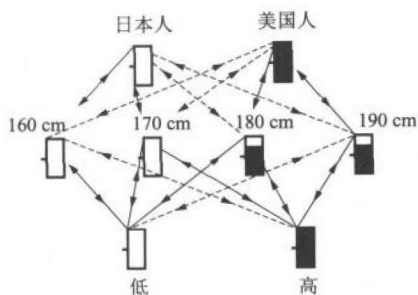


图 4.7 “个子高的美国人”的 CFS

这样得到的 CFS 中若使代表‘个子高’和‘美国人’的节点激活，如图 4.7 所示，中位层中激活度分布就被表示出来。这表现了依存于“美国人”这一状况的模糊集合“个子高”的隶属函数。这个例子中分散性表示了对人种的身高尺寸的 CFS 和关于身高及身高尺寸的 CFS 实现了图 1.9 中说明的依存状况的意思表示。

4.5 模糊度和不确定性

现实世界的信号中含有各种各样的模糊性，我们已经叙述了将其作为集合中的模糊性来处理的基本方法。把‘集合中的模糊性’称为模糊度(fuzziness)。

另一方面，从不确定性的角度，对现实世界的模糊性信号的表示，提出了模糊量度的概念^[8]。模糊量度，用主观的量度量现实世界的模糊性信号。而且，通过用模糊积分综合评价模糊性信号，可以适用于故障检测等信号处理方面。

从模糊度的角度看过去的方法，存在模糊性扩大的弱点，不适用

于信号的综合评价。而现在，通过与神经元的融合的一般化，模糊性的控制成为可能，从模糊度角度来看可以适用于信号评价。

因此，本书采用了模糊神经网络 (fuzzy neural network) 的信号评价方法，并表示了其应用方法。第 7 章将叙述含有多模糊性的手语识别的例子，并说明有关模糊度观点的智能信号评价方法。

第5章

模糊推理和模糊 信号处理



- 5.1 模糊推理的思想
- 5.2 模糊信号处理中的推理
- 5.3 应用神经网络的模糊推理
- 5.4 模糊信号处理和模糊控制

模糊推理，是把模糊集合中的模糊性的概念进行扩展，能够根据模糊规则来处理逻辑性的模糊性。而且，它还可以将神经网络应用于模糊推理中。模糊信号处理是把模糊推理应用于信号处理。本章将详细说明模糊推理和模糊信号处理。

5.1 模糊推理的思想

把集合中的模糊性进行扩张，表示处理逻辑中的模糊性，就是模糊推理(fuzzy inference)。模糊推理被认为是根据模糊规则进行的近似推理，是多值逻辑的延伸上的推理。

我们已经知道有很多推理方法。这里将叙述根据模糊规则把现实世界的数值信号由模糊规则评价，实施向人能够理解的符号和其性质程度变换的模糊信号处理的推理方法。

也就是说，对以前用得很多的模糊推理以及神经网络和模糊系统相融合的模糊神经网络中的推理进行解说。而且在通过将信号反馈来使用的情况下，即使是在数字信号中，模糊控制系统的概念也是有用的。因此，对于稳定性和规则的精练等模糊控制系统中的模糊推理等诸问题进行讨论。

5.2 模糊信号处理中的推理

在对现实世界的数值信息进行评价并给出符号所代表的意思内容以及识别其程度的过程中，人类表现感觉综合等(参见图 1.8)的内在层次性的意思，以及实施基于这种表示的实时认知循环。模糊信号处理，是把人的观察中有控制论方面的信号处理在工学上进行简化，

以实现为目的。因此，根据模糊规则，可有效使用模糊推理。

模糊推理有多种形式，这里从实时性等观点出发介绍适用于模糊信号处理的推理方法，即模糊控制中使用的模糊规则表示及其推理方法。

5.2.1 模糊规则

重视实时性的模糊控制中的推理，把专家的主观模糊性，比如，“正的一侧很大 (PB)”，…，“零附近 (ZO)”，…，“负的一侧很大 (NB)”等用图 4.2 所示的隶属函数进行表示，用下述规则表示：

$$\text{规则 1: if}(x_1 \text{ is PB}) \& (x_2 \text{ is ZO}) \text{ then } y \text{ is PM} \quad (5.1)$$

$$\text{规则 2: if}(x_1 \text{ is ZO}) \& (x_2 \text{ is PB}) \text{ then } y \text{ is PS} \quad (5.2)$$

这样，模糊规则中，把专家知识记述为由 if 组成的前件和由 then 组成的后件。规则 1 表示的是，如果输入值 x_1 为正的一侧很大，并且若输入 x_2 在零附近，则输出值为在正的一侧稍大，这样的模糊命题。

规则中 x_1, x_2 表示输入， y 表示输出。PB, PM, PS, ZO 分别表示 positive big, positive medium, positive small, zero，的意思，是表示专家的主观模糊性的模糊集合，根据隶属函数对其进行定义。

在高木-营野的模糊模型^[2]的知识表示中

$$\text{规则 3: if}(x_1 \text{ is PB}) \& (x_2 \text{ is ZO}) \text{ then } y = f_1(x) \quad (5.3)$$

$$\text{规则 4: if}(x_1 \text{ is ZO}) \& (x_2 \text{ is PB}) \text{ then } y = f_2(x) \quad (5.4)$$

像规则 3, 4 那样，由 then 组成的后件不是模糊集合，而是 $f_1(x), f_2(x)$ ，这样的输入输出关系式。因此表述能力被提高，在模型化和模糊适应控制中得到了广泛应用。

(1) 修饰算子对于前面所述的模糊集合，下式中有下划线的为修饰算子。

$$\text{规则 5: if } x_1 \text{ is } \underline{\text{not}} \text{ PB then } y \text{ is PB} \quad (5.5)$$

$$\text{规则 6: if } x_1 \text{ is very PB then } y \text{ is PB} \quad (5.6)$$

其中规则 5 中的 not 规则 6 中的 very 修饰算子，与模糊集合 PB 有

关。这些修饰算子通过在隶属函数中进行一定的运算，可以进行同样处理。除此之外的修饰算子还有 more, less, plus, minus, highly, weak 等各种。借助于这些修饰算子，模糊规则变得容易理解。

5.2.2 模糊推理

下面将对模糊控制中用到的基本推理方法予以介绍。在下述的模糊规则 7, 8 中，将观测值 $x_{01}=0.7, x_{02}=0.2$ 分别输入给 x_1, x_2 。

规则 7: if (x_1 is PM) & (x_2 is ZO) then y is ZO (5.7)

规则 8: if (x_1 is ZO) & (x_2 is PM) then y is NM (5.8)

式 (5.7) 式 (5.8) 的规则 7 和规则 8 对应于 $x_1 = x_{01}, x_2 = x_{02}$ 的隶属函数 $g_{07}(y), g_{08}(y)$ 用下式求得。

$$g_{07}(y) = (\mu_{PM}(x_{01}) \Delta \mu_{ZO}(x_{02})) \bigcirc \mu_{ZO}(y) \quad (5.9)$$

$$g_{08}(y) = (\mu_{ZO}(x_{01}) \Delta \mu_{PM}(x_{02})) \bigcirc \mu_{NM}(y) \quad (5.10)$$

式中 Δ 表示模糊逻辑运算 AND (min)。 \bigcirc 称为一般化合成规则的 T —范数运算符，表示描述与 AND 一般化的合成规则有关的某种关系的运算符。通过适当选择 T —范数运算符，实现与状况对应的推理。

这里，对 T —范数运算符进行 \cdot (乘算) 时，为

$$g_{07}(y) = w_{07} \cdot \mu_{ZO}(y) \quad (5.11)$$

$$g_{08}(y) = w_{08} \cdot \mu_{NM}(y) \quad (5.12)$$

式中 w_{07}, w_{08} 表示隶属度为

$$w_{07} = \mu_{PM}(x_{01}) \Delta \mu_{ZO}(x_{02}) = 0.7 \quad (5.13)$$

$$w_{08} = \mu_{ZO}(x_{01}) \Delta \mu_{PM}(x_{02}) = 0.2 \quad (5.14)$$

作为最后的结论，根据

$$g_0(y) = \bigvee_{i=7,8} g_{0i}(y) \quad (5.15)$$

求得输出的隶属函数。式中的 \bigvee 表示模糊逻辑演算 OR (max)。图 5.1 表示了这一推理过程。模糊推理中，有关系的全部规则，与其确定度成比例，并行完成。

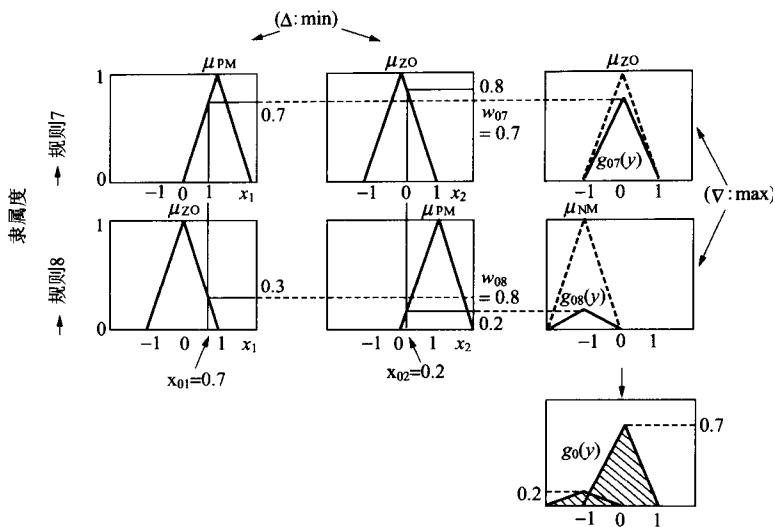


图 5.1 模糊推理过程

(1) T -范数运算符为了使模糊推理的结果与专家的推理相一致，作为模糊规则的前件和后件的合成规则，使用把 \min 运算一般化的 T -范数运算符 \circ 的系统。

$$x \circ y = \begin{cases} x \Delta y, & t=0 \\ x \cdot y, & t=1 \\ \max(0, x+y-1), & t=\infty \\ \log\left(1 + \frac{(t^x-1)(t^y-1)}{t-1}\right), & 0 < t < 1, 1 < t < \infty \end{cases}$$

(5.16)

式中 $0 \leq x \leq 1, 0 \leq y \leq 1$ 。

根据 T -范数运算符的参数 t 变化，

Mamadani 推理... $t=0$

Larsen 推理... $t=1$

界限积的推理... $t=\infty$

各种推理成为可能，因此能够实现与状况对应的推理。其隶属函数的变化如图 5.2 所示。

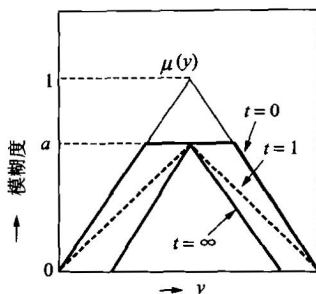


图 5.2 T-范数运算子的隶属函数的变化

(2) 输出的合成方法在模糊控制的情况下，必须从作为推理结果的输出的模糊集合 $g_0(y)$ 合成求出实际的操作量 u^* 。这被称为非模糊化。将前述的模糊集 $g_0(y)$ 在 y 轴上进行 n 个离散化用 $y_i (i=1, 2, \dots, n)$ 表示时，作为操作量 u^* 的合成方法

$$u^* = \frac{\sum_{i=1}^n y_i \cdot g_0(y_i)}{\sum_{i=1}^n g_0(y_i)} \quad (5.17)$$

$$u^* = \max\{g_0(y_i)\} \quad i=1, 2, \dots, n \quad (5.18)$$

经常用到。

5.3 应用神经网络的模糊推理

与以前的 AI 相比，模糊系统允许模糊性的表示，具有积极利用模糊性的特征，与神经网络相比有明确表示知识的特征；另一方面在模

糊系统的知识表示中，除了模糊理论外，有必要采用在 AI 中使用的被称为框架的结构。

而且，对于学习结构，也有必要采用 AI 及神经网络技术，单独的模糊系统有很多不完备的地方。

模糊系统与神经网络技术相融合的研究始于 1990 年前后，并且其数量飞速增长。本节中，根据模糊系统和神经网络的共同点和不同点，叙述其相融合的优点和模糊神经网络的分类，并说明引入神经网络的模糊推理法在模糊神经网络分类中的作用和特征。

5.3.1 模糊系统角度上的神经网络

如何求隶属值来构成模糊集合，或者怎样求推理的规则，可以说是知识获得的根本课题。为了解决这一课题，期待提高神经网络的学习能力，同时以下所示几个方面的问题中就是要试图将模糊系统和神经网络相融合的研究吸引了不少学者。

隶属函数的生成；

逻辑操作的生成，逻辑操作的替换；

推理规则的生成；

从获得的知识的推理中，作为改善以前模糊推理的缺点“模糊性的膨胀（推理时模糊性渐渐扩大的现象）”的方法，神经网络的联想功能（对模糊性的事件，回想最与条件相匹配的存储功能）很重要，因此有下面的融合研究。

过去形式的推理方法的改善和新的推理方法的实现。

作为这一推理的延伸，可以向不局限于过去规则形式的新的模糊知识表示的方向进行扩展。

5.3.2 神经网络角度上的模糊系统

从神经网络角度看，以下的研究正在开展过程中。由于在神经网络上知识被分散隐含，因而存在不能明确看到的问题。因此，就需要考虑

用模糊知识表示来解释神经网络的学习结果，用模糊推理规则形式把获得的知识作为初始知识来进行神经网络的学习，以实现快速化。

还有，把模糊推理规则和推理结构引入神经网络，试图对网络本身进行结构化。或者把神经元的激活值分布直接与模糊知识表示相联结，来增强模糊性知识表示。

综上可归纳为：

知识的解释

作为初始知识的利用

处理机构的结构化

模糊性知识的表示

对模糊系统的认识正在不断扩大。

5.3.3 模糊神经网络的分类和推理方法概要

(1) 按形式分类在模糊系统与神经网络结合构成的系统例子中，从单一的两个系统的连接到融合等，已经出现了很多形式^[3]。在这些构成例子的分类中，如图 5.3 所示，以下四种按形式分类的方式最为人们所知。

(a) 模糊系统与神经网络的连结^[4]

(b) 模糊系统对神经网络起辅助作用^[5]

(c) 神经网络对模糊系统起辅助作用^[6]

(d) 模糊系统与神经网络完全融合。

上述(a)~(c)为两个系统相分离的形式结合，(d)由于两个系统密切结合不能分离。特别是(d)的形式，由于具有几个突出优点，研究正在深入进行。

以后，我们将详细介绍这些形式。

(2) 按结构分类在模糊领域，用最一般的规则来表示知识，以实现推理规则和推理过程，在模糊规则明确的情况下，根据与神经网络替换的部分，可以分为图 5.4 所示的四类结构。

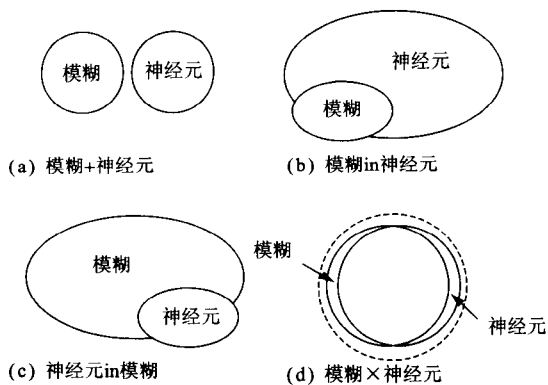


图 5.3 按形式分类

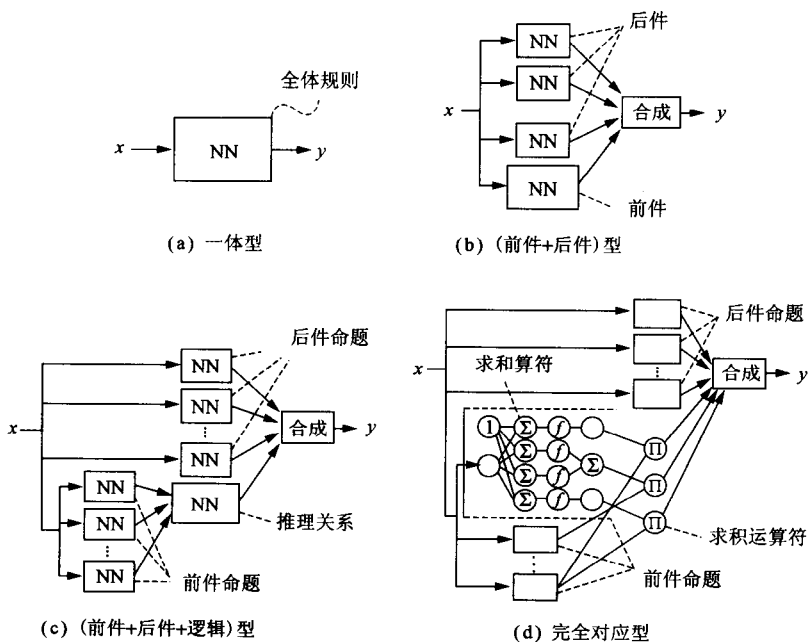


图 5.4 按结构分类

由于这种形式是用非常一般的模糊规则的融合形式，所以有如以下所示的研究^[3]

- (a) 一体型^[7];
- (b) (前件 + 后件)型^[8];
- (c) (前件 + 后件 + 逻辑)型^[9];
- (d) 完全对应型^[10,11]

这里 (c) 的结构与模糊知识表示相近，构成规则的要素命题通过个别来实现。因此，由于连同命题可以个别进行学习，学习时间的快速化和学习后知识的提取比较容易。由于表示逻辑关系的网络独立存在，可以实现弥补过去的模糊推理弱点的新的推理方法。后面将会举例说明模糊联想存储。

(3) 按知识表示分类在按结构进行的分类中，虽然是以称为规则的知识表示为基础，但也有以下的表示形式，它们是把模糊知识的表示直接在神经网络上展开，立足于本来知识带有不确切性的识别，以对用神经网络的模糊性知识进行表示为目的。

动态知识的表示^[12,13];

与神经元水平的模糊性知识的融合^[14~17];

概念模糊集合(CFS)^[18]

模糊集合的隶属值，通常用算术函数计算，但在 的方法中如果全体集合不是数值的情况下计算相对困难。概念模糊集合(CFS)中，模糊集合为意思表示的形式，所以联想存储上的激活值的分布与模糊集合相一致。概念的意思由其他的概念以及其激活值来表示，其激活值通过联想存储来计算。

因此，不必通过算术函数计算隶属值。此外，推理中根据激活值的传播，实现动态表示成为可能。

以上，从三个不同的角度，展示了对模糊神经网络进行分类的方法。模糊系统和神经网络，在输出信号限定在 $[0,1]$ 区间的情况下，场信号处理状态空间由于可以在能够用 n 维超立方体表示的意思上视为相同，从

而能够实现将它们融合的模糊神经网络，而不会出现不相容感。

特别是，由于两者融合，优势增强，具有预防学习时网络的暗箱化，实现人类易于理解的学习，并能够预防推理时模糊性增大的联想推理的实现等特征。

下节将要说明的控制模糊性的联想推理，并在 5.4 节介绍的与人的亲和性高的学习。模糊神经网络正如第 7 章中介绍的那样，规则的生成和构思，可以说对于成为今后重要的新的信号处理，是十分有效的系统。

5.3.4 模糊联想推理

在以前的模糊推理中，推理结果中的模糊性程度有增加的趋势。因此，对于将数值信号进行综合评价，向人能够理解的信号变换的模糊信号处理来说，就不能适应了。

但是，通过应用将神经网络的联想存储进行融合的联想推理，就能够控制模糊性，可以有效地应用到信号处理中。模糊联想推理系统的基本构成要素是神经网络的双向联想存储（BAM）（参见 3.3 节）。本小节将叙述模糊联想推理系统的构成过程，联想推理过程和联想推理的特征等^[3]

(1) 双向联想存储 双向联想存储（BAM）是如图 5.5 所示的用二层结构的网络。各层中有节点并且节点之间结合的权值由联想矩阵 $M \in R^{n \times p}$ 表示。

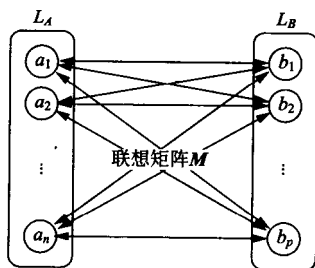


图 5.5 双向联想存储

此联想矩阵对于 L_A, L_B 层 若对存储模式对 $A \in R^n, B \in R^p$ 进行存储, 即使对 L_A 层施加含有噪声的 $A' \in R^n$, 由于层间有反响动作, B 仍可被回想。相反, 对 L_B 层施加含有噪声的 $B' \in R^p, A$ 也可被回想。这里, BAM 的联想按式 (5.19) 式 (5.20) 进行。

$$A(t+1) = \phi(M^T B(t)) \quad (5.19)$$

$$B(t) = \phi(MA(t)) \quad (5.20)$$

式中 $A(t) = [a_1(t), a_2(t), \dots, a_n(t)]^T, B(t) = [b_1(t), b_2(t), \dots, b_p(t)]^T$ 为步骤 t 中的各节点的激活值向量, 在 $[0, 1]$ 中取值。 $\phi(\cdot)$ 为各节点的函数, 用式 (5.21) 所示的双弯曲函数求得。

$$y_i = \frac{1}{1 + \exp(-\lambda \cdot net_i)} \quad (5.21)$$

式中, y_i 为节点 i 的输出值, net_i 为节点 i 中被赋予权值的输入值的总和。 λ 为决定函数梯度的常数。

联想矩阵 M 若要存储的模式对为 $(A_1, B_1), \dots, (A_m, B_m)$ 则由下式求得。

$$M = \sum_{i=1}^m \beta B_i A_i^T \quad (5.22)$$

式中的 β 为联想系数。

在 Kosko 的方法^[19]中, 求联想矩阵时, 使用了把 $\{0, 1\}$ 模式对进行双向变换的模式对。但是, 用这一方法求得的联想矩阵, 由于有可能引起网络的饱和状态, 必须加以调整。因此, 要用到通过联想矩阵的正规化求得的正规化联想矩阵。

(2) 联想推理系统的构成过程式 (5.23)~式 (5.25) 为简单的模糊规则的例子 图 5.6 表示其隶属函数。

规则 9: if 身高 is 高 体重 is 重 腰围 is 粗, then A (5.23)

规则 10: if 身高 is 一般 体重 is 一般 腰围 is 一般, then B (5.24)

规则 11: if 身高 is 矮, 体重 is 轻 腰围 is 细, then C (5.25)

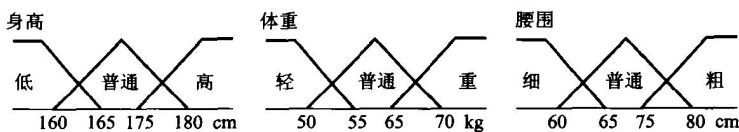


图 5.6 隶属函数

以这个规则为例说明推理部分的构成过程。由式 (5.23) ~ 式 (5.25) 的模糊规则构成如图 5.7 所示的网络。

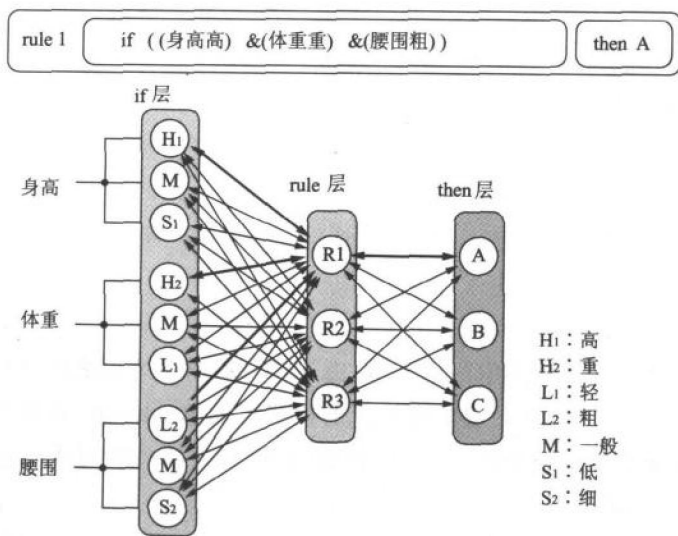


图 5.7 模糊规则

关于层的设定，把代表模糊规则前件的层作为 if 层(把要素称为 X_i)，把代表后件的层作为 then 层(把要素称为 Y_i)。if 层内的节点代表前件的模糊集合(隶属函数)，then 层内的节点代表后件的模糊集合(隶属函数)，或者代表输入输出函数。

前件和后件的相关，由于超过了 BAM 能够记忆的限制，所以设定

一个节点代表一个规则的 rule 层(把要素称为 R_i)。因此,模糊规则能够通过 在 if 层和 rule 层之间(联想矩阵 M_x),rule 层和 then 层之间(联想矩阵 M_y) 构成 BAM 来表示。

在 rule 层,为了防止模糊性的增大,设定具有抑制其他节点激活作用的调整结合(联想矩阵 M_r)。所谓调整结合,是同一层内的任意节点中,对自己用正的权值,对其他节点用负的权值进行结合。

下面,为了求各个层之间的联想矩阵,首先把模糊规则的相关替换成式(5.26)所示的二值数据。

	身高	身高	身高	体重	体重	体重	腰围	腰围	腰围
	高	一般	矮	重	一般	轻	粗	一般	细
$X_1[$	1	0	0	1	0	0	1	0	0]
$X_2[$	0	1	0	0	1	0	0	1	0]
$X_3[$	0	0	1	0	0	1	0	0	1]

(5.26a)

	R1	R2	R3		A	B	C
$\leftrightarrow R_1[$	1	0	0]	$\leftrightarrow Y_1[$	1	0	0]
$\leftrightarrow R_2[$	0	1	0]	$\leftrightarrow Y_2[$	0	1	0]
$\leftrightarrow R_3[$	0	0	1]	$\leftrightarrow Y_3[$	0	0	1]

(5.26b)

作为应该存储其值的模式对,由式(5.22)求各个层之间的联想矩阵。不过,不是根据用双向变换了的模式对的 Kosko 法,而是在推理时应用正规化联想矩阵。这时,由式(5.27)式(5.28)可求得正规化联想矩阵 M_r 。

$$M_r = a(M + B) \quad (5.27)$$

这里

$$a = \frac{2[\max\{m_{ij}\} - \min\{m_{ij}\}]}{3} \quad (5.28a)$$

$$B = \begin{bmatrix} -c & -c & \cdots & -c \\ -c & -c & \cdots & -c \\ \vdots & \vdots & \ddots & \vdots \\ -c & -c & \cdots & -c \end{bmatrix} \quad (5.28b)$$

$$c = \frac{1}{mn} \sum_{i,j} m_{ij} \quad (5.28c)$$

式中 $M \in R^{m \times n}$, $M_c \in R^{m \times n}$, $B \in R^{m \times n}$, m_{ij} 表示 M 的 (i, j) 要素。

(3) 联想推理的特征联想推理存在这样的性质, 即由于用双向联想存储构成网络, 所以具有将规则前件命题的激活值向规则后件命题的激活值传播的自下而上的处理, 和将规则后件命题的激活值向规则前件命题的激活值传播的自上而下的处理相融合的传播处理的双向性。这样的自下而上处理和自上而下处理的融合, 在这里称为“直感”。根据此直感的联想推理, 可进行满足 Klir 的模糊集合的公理 1~10(参见 4.3 节)的近似推理。

还有, 对于激活值的传播次数(联想步骤数) 在希望得到更加明确的推理结果时增多, 在希望保留模糊性的推理结果时减少, 因而具有能够控制模糊性的特征。

作为这一特征的表现, 表 5.1 表示了联想推理和式(5.22) 的模糊规则中的过去模糊推理的推理结果。与 A 百分之百匹配的值, 组合输入的情况下, 模糊推理和(联想步骤设定为 4 的)联想推理都输出了适当的结果。

表 5.1 推理结果

推理法	身高	体重	胸围	A	B	C
模糊推理	180.00	70.00	80.00	1.000	0.000	0.000
	180.00	45.00	80.00	0.000	0.000	0.000
联想推理(第 4 步)	180.00	70.00	80.00	0.965	0.017	0.017
	180.00	45.00	80.00	0.901	0.049	0.049
联想推理(第 3 步)	180.00	70.00	80.00	0.838	0.000	0.162
	180.00	45.00	80.00	0.683	0.081	0.236

但是 只含有一个与 A 不匹配的值的组合输入的情况下, 联想推理输出了恰当的结果, 而模糊推理则没有输出满意的结果。

而且,设定联想步骤为 3 步的联想推理结果,与设定联想步骤为 4 步时相比,输出了保留有模糊性的结果。用联想推理是能够控制模糊性的。

如上所述,通过推理对输入信号进行综合评价,向人能够理解的信号变换时,通过调整联想步骤,可以回避过去的模糊性增大的问题。在过去的推理方法中,即使只有输入信号的一部分有模糊性,推理结果的全体将变得模糊。因此,层次性知识的推理中,就不能够得到有效的推理结果。

通过使用联想推理,“多段推理”成为可能,可以有效地进行信号的综合评价。相关例子将在第 7 章的手语识别中予以介绍。

5.4 模糊信号处理和模糊控制

在模糊信号处理有效的应用例子中有考虑反馈系统的模糊控制。模糊控制可以认为是为了评价控制对象的状况而把模糊信号进行内在的构成。本节将叙述有反馈系统的模糊控制器的稳定性以及模糊神经网络的学习性。

5.4.1 模糊控制的系统构成和稳定性

如图 5.8 所示,最一般的模糊控制器为,根据模糊规则输入来自被控对象 P 的数值信号,对被控对象的内部状况进行评价,输出对应于其状况的操作量。

被控对象的变动比较大,以前的线性控制器及滤波器难以对应时,而通过应用模糊控制器就能够与变动状况对应。通常,模糊控制系统,由于有反馈循环结构,模糊规则的设计者必须考虑系统的稳定性。这与由模糊控制器构成的数字滤波器的情况相同。

图 5.8 的结构是式(5.1)所示的规则形式。由式(5.3)所示的高木、菅野的模糊模型的规则形式中的结构,表示在图 5.9 中。本节为

了对它们予以区别，把图 5.8 所示形式的模糊控制器称为“直接式模糊控制器”，而把图 5.9 所示结构的形式称为“模糊模型式控制器”。

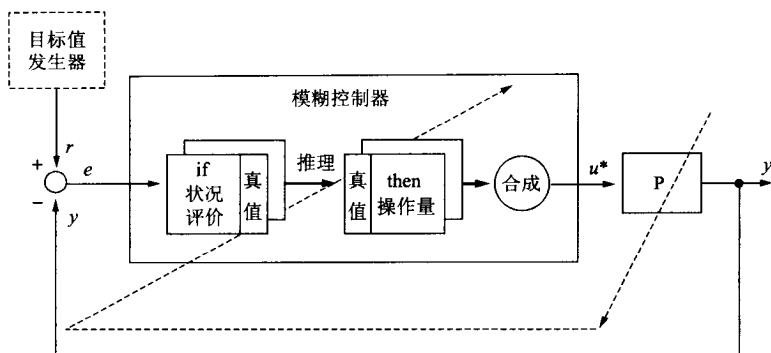


图 5.8 模糊控制系统结构

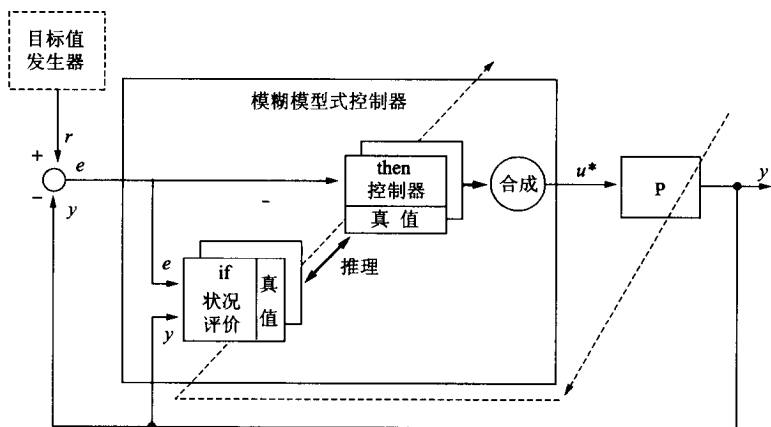


图 5.9 基于模糊模型的控制器结构

直接式模糊控制器的构成中，由于推理循环是操作量的输出循环，就特别需要进行快速处理。因此，模糊推理的硬件化¹⁾的实现就非

1) 参看本系列书中的《VLSI和数字信号处理》一书。

常重要。

在模糊模型式控制器的构成中，由于是根据推理来选择输出操作量的控制器，如果是获得控制对象的变动状况的程度循环，推理循环就很充分。所以，用软件就可以进行推理处理，这在很多的智能系统中得到了应用。

在这些控制系统中，考虑被控对象的变动，通过使与其对应的操作量发生变化，能够实现有效的适应操作，这是其优点之一。另一方面，这样的动态系统，却存在着稳定性方面的问题。在直接式模糊控制器的控制系统中，尽管将变动的状况用语言形式表现，仍能够进行稳定性^[22~24]方面的讨论。

通过应用模糊模型式控制器，可以在近似于直接式模糊控制器的同时，讨论以前系统理论的延伸的稳定性。从李亚普诺夫的稳定理论的观点看，也可以探讨本系统的稳定性。本节根据多重线性系统的模型化的观点，叙述讨论稳定性的方法。图 5.10 是本方法的模糊系统的一个模型。

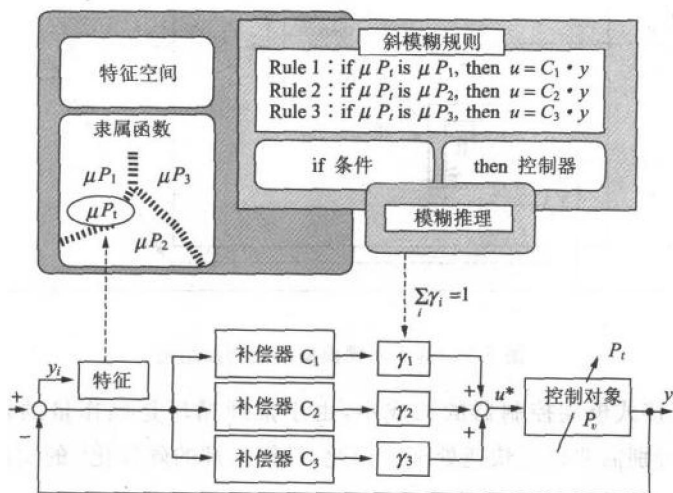


图 5.10 模糊系统及其模型化

图 5.10 中, $C_i (i=1, 2, 3)$ 是与控制对象的状况对应的线性补偿器, 即控制器。根据模糊推理对控制对象 P 的变动进行评价, 为了使时刻 t 的变动状况与近似的线性控制对象 P_i 相对应, 进行了多重线性补偿器的凸结合 (合计为 1 的权值 $\gamma_i \geq 0$ 的合成结合)。在这个例子中, 控制对象 P 的变动, 假定可以用 P_1, P_2, P_3 三个线性模型代表。而且带有与这三个模型对应的三个模糊规则。

讨论本模型的稳定性时, 必须考虑图 5.11 所示的控制对象的变动模型的“同时稳定化问题”和图 5.12 所示的线性补偿器的合成模型的“凸结合型多重补偿器问题”。

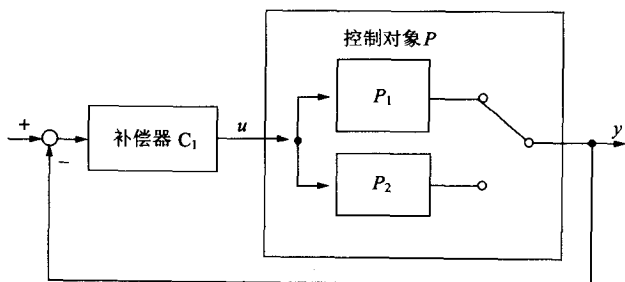


图 5.11 同时稳定化问题

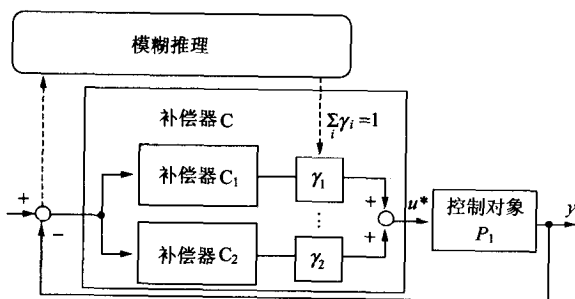


图 5.12 凸结合型多重补偿器问题

本模型中，尽管得到满足稳定性的一般解很困难，但是对象明确时，通过应用这个模型化能够导出条件。满足稳定性条件的导出方法在附录中叙述，而较简单的例子在 5.4.3 节中说明。

5.4.2 模糊神经网络的学习性

上一节展示了模糊系统的模型化的一个方法，这里，讨论关于把神经网络导入这个模型时的学习性。图 5.4 的模糊神经网络中，在图 (d) 完全对应型的情况下与层次型神经网络相同，可以通过利用教师信号由反向传播来进行规则的学习^[10,11]。

如图 5.13 所示，也有应用 LVQ 参见 2.5 节），根据输入信号的特征实施模糊分割，生成评价信号的规则前件的隶属函数这样的方法^[20]。

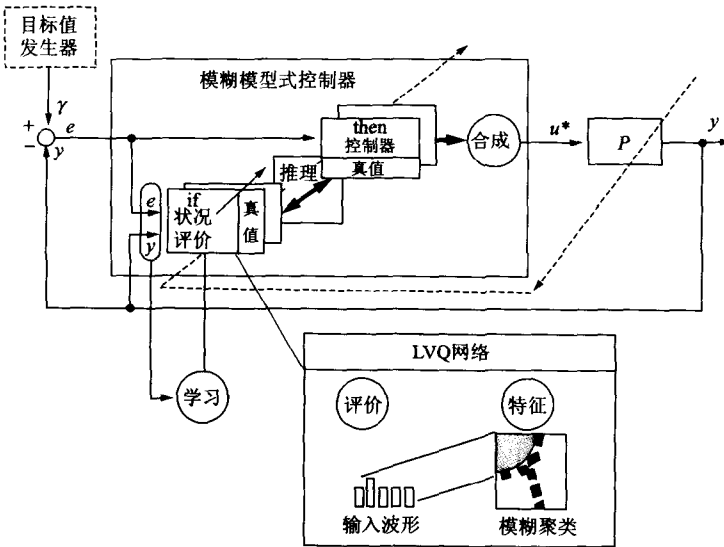


图 5.13 考虑规则前件学习性的模糊控制器结构

这一方法，根据输入信号的特征自律生成分类评价的隶属函数，从实际的数值信号观测系统的变动，对于与亲和性高的语言性表示的联结，是有效的。所以可以说，构成了某种模糊信号处理。

与语言性表示的系统变动相对应，调整操作量的规则后件部的学习方法见图 5.14。这里，对应各个状态的规则后件的控制器，由层次型神经网络构成，通过教师信号的平方误差最小化的反向传播等学习方法进行简化。

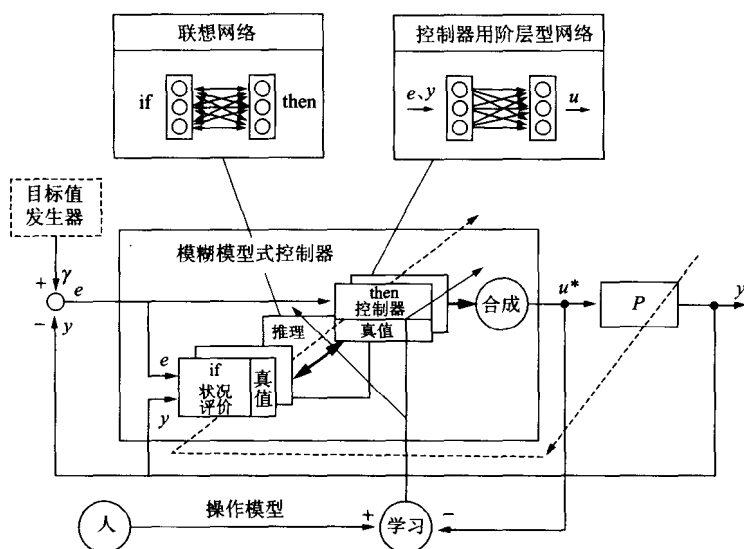


图 5.14 考虑规则后件学习性的模糊控制器结构

这种情况，与用语言性表示的规则前件的各个状态相对应，为了分别地对后件的各网络块进行学习，在能够快速学习的同时，能够减少各块的存储容量，使网络更简单。

还有像在 5.3.4 节中介绍的那样，由于模糊联想推理在对输入信号的评价方面表现出色，在应用模糊联想推理时，由神经网络的联

想存储可以实现规则的逻辑关系。这时，伴随着规则后件网络的简化，与前件的逻辑关系由 Hebb 学习进行强化。下面对这些学习过程进行举例说明。

5.4.3 模糊控制中的学习性和稳定性

把前述有关学习性和稳定性的结果应用于圆盘型直升机的悬停操纵模型，来证实其有效性。下面说明用于悬停操纵的知识提取方法和知识简化方法。

(1) 基于模糊模型的语言性表示和学习性及稳定性模糊模型式的知识表示中，操纵的知识用前件 (if) (状况 (比如直升机的飞行状况 高空飞行，低空飞行等)) 和后件 (then) (操作模型 (控制器)) 来表示。

而且，使用由模糊规则得到的关系，通过互联型神经网络的联想存储实现这些“状况”和“操作模型”。模糊模型用某种函数来表示各规则的模糊命题的“状况”和“操作模型”。其结构可以用图 5.10 的模型表示。

作为模糊模型学习性说明的例子，采用图 5.15 所示的三个规则的直升机操纵的操作模型 f_1, f_2, f_3 的结构。 C_{ui} 称为关联度，表示其规则 R_i 中的理想操作模型 f_i^* 和现状操作模型 f_i 的关联。通过学习操作模型 f_i 被简化 如果接近理想 则关联度 C_{ui} 接近于 1。

这样，关联度作为联想矩阵权值，表示了概念之间的关联。联想推理时，由于趋近输入条件，被很好简化了的高关联度 C_{ui} 的操作模型通过联想进行回想，对操作量的确定起到了重要作用。根据这个贡献度 γ_i 通过对控制器 f_i 的参数的合成，实现适应动作。

这个学习适应功能，就变得与人的操作知识有关的学习性和适应性相近。

关于稳定性，根据控制器 f_i 和控制对象 P_i 的同时稳定化问题来求条件并在 f_i 的学习时使用。

(2) 实验系统概要图 5.16 是模型圆盘型直升机操纵的实验系

统概要。操纵者目测机体的位置，通过前面叙述的模糊系统，使用无线电控制发射机的操纵杆，实现一系列操纵。

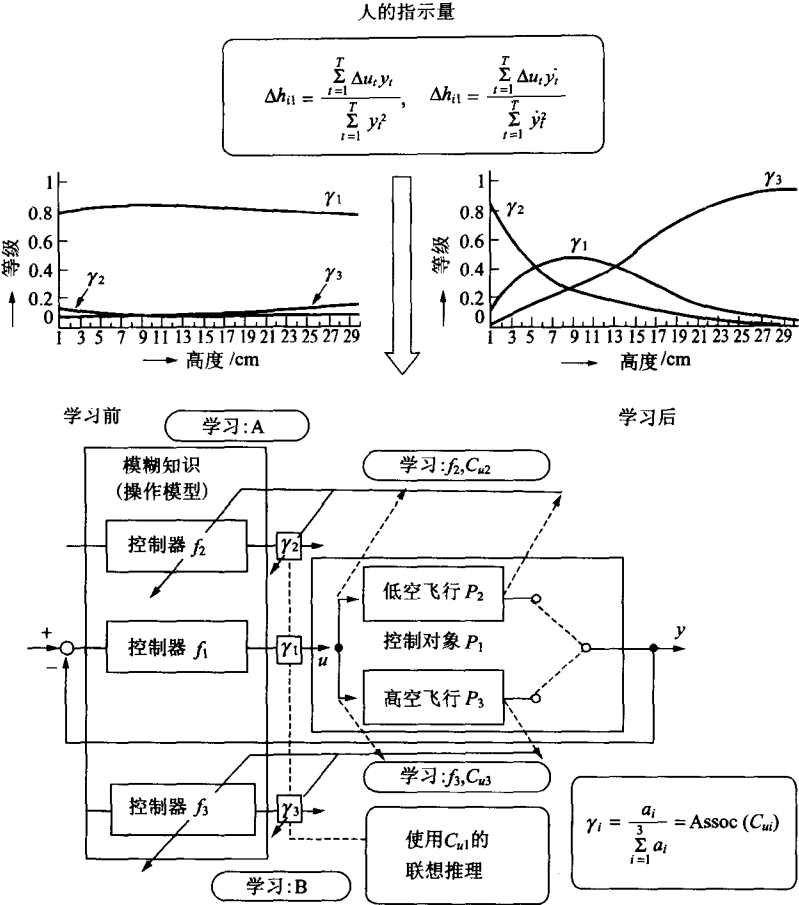


图 5.15 模糊控制器的学习性

本系统中，使用相当于操纵者眼睛的两个二维 PSD (Position Sensitive Light Detector) 摄像机。控制器把这两个摄像机的二维位

置数据三维化，根据操纵者的模糊规则进行联想推理，确定操作量 u^* ，实现在线悬停操纵。

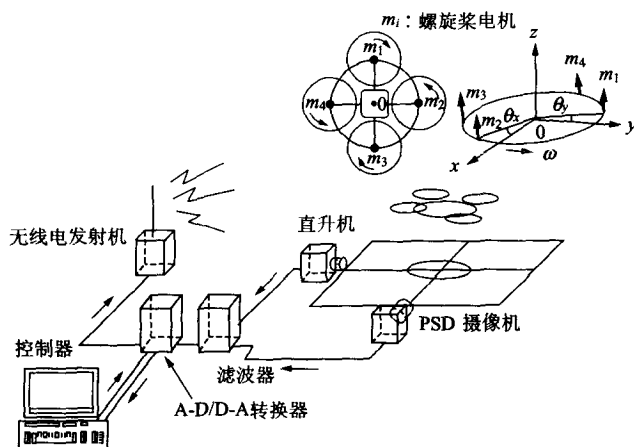


图 5.16 模型圆盘型直升机操纵实验系统概要

(3) 直升机的特性和操纵知识的提取操纵者知道，在高空飞行或低空飞行，机体有较大倾斜等情况下，特性也会有很大变化。它就要了解最大的影响要素，着眼于高度的特性变化。根据模糊模型把“高空飞行”和“低空飞行”等的高度的特性变化进行模型化并根据模糊模型的语言性表示，提取操作模型。

(4) 直升机的模糊模型和语言性表示模型直升机是图 5.16 上部所示的结构，使用四个直流电机的电动风叶 $m_i (i=1, 2, 3, 4)$ 在三维空间飞行。

操作系统和普通的直升机相同，有 加速（上升、下降：操作 z 轴）； 偏转（回转：操作 ω ）； 上下颠簸（前后倾斜，移动： θ_y 的倾角和操作 y 轴）；④左右摇摆（左右倾斜，移动： θ_x 的倾角和操作 x 轴）4 个系统。

这个圆盘型直升机由于是对称结构，上下颠簸和左右摇摆有相同特性。下面只取上下颠簸操作进行说明。关于左右摇摆和偏转，也可以应用与上下颠簸同样的方法。

依据操纵者对直升机动态特性的知识，实现表示高空飞行和低空飞行等不同状况的参数变动的模糊模型。根据各种已经进行的鉴定试验结果，用式(5.29)所示的四次传递函数 P_v 来近似直升机系统：

$$P_v = \frac{k_v e_0}{s^4 + k_v d_3 s^3 + k_v d_2 s^2} \quad (5.29)$$

式中 k_v 为变动参数 ($k_3 \leq k_v \leq k_2$)， e_0, d_2, d_3, k_0 为正的系数。

低空飞行时由于来自地面的地面效应的作用，操纵的感觉度增高 ($k_v = k_2$ 状况)，而且来自地面的湍流，使干扰增加；高空飞行时没有地面效应，操纵的感觉度变低 ($k_0 = k_3$ 状况)。表 5.2 中表示各系数值。

表 5.2 直升机系统的各系数

$e_0 = 652.4$	$k_3 \leq k_v \leq k_2$
$d_2 = 313.7$	$d_3 = 15.68$
$k_2 = 0.052$	$k_3 = 0.029$

作为直升机的特性，操纵者将发现这二个特征模型。把这些特征的低空模型取为 P_2 ，高空模型取为 P_3 。应用模糊模型，变动的被控对象模型 P_1 按下式定义：

$$P_1 = \alpha_2 P_2 + \alpha_3 P_3 \quad (5.30)$$

上式表示两个参照模型 P_2, P_3 的参数值的内插。这里， $\alpha_2 + \alpha_3 = 1, \alpha_2 \geq 0, \alpha_3 \geq 0$ 。而且 $\alpha_i (i=2,3)$ 为与高度 h 中各个参照模型对应的隶属值表示为图 5.17 的形状。下面是模型 P_2, P_3 的提取方法。

(5) 有关操纵的初始知识的提取对于有高空和低空两个代表性参照模型的直升机，操纵者首先选用都可操纵的两个模型中有鲁棒性的操作模型进行操纵。与全域模型 P_1 对应的鲁棒性的操作模型为

f_1 。操纵者重复训练 来简化 f_1 得到对应于各状况(高空飞行, 低空飞行)的专用操作模型。与低空模型 P_2 , 高空模型 P_3 对应的操作模型分别为 f_2, f_3 。

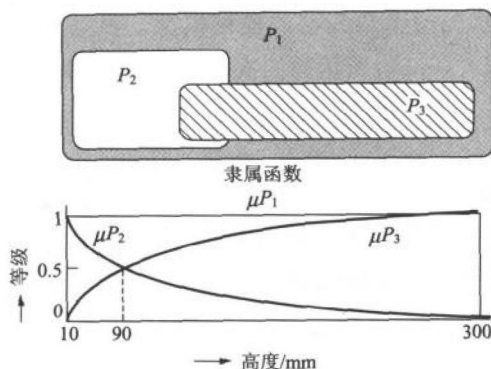


图 5.17 模型直升机模糊模型的隶属函数

把以上有关操纵的知识的概要, 用模糊模型的形式表示如下:

$$\text{规则 1: if } h \text{ is } \mu P_1, \text{ then } u = f_1(y, r) \text{ with } C_{u1} \quad (5.31)$$

$$\text{规则 2: if } h \text{ is } \mu P_2, \text{ then } u = f_2(y, r) \text{ with } C_{u2} \quad (5.32)$$

$$\text{规则 3: if } h \text{ is } \mu P_3, \text{ then } u = f_3(y, r) \text{ with } C_{u3} \quad (5.33)$$

式中 h 为向规则输入的机体的高度, y 为距机体目标点的变位 u 为操作量, μP_i 为与各参照模型 P_i 对应的模糊集合, 代表 μP_1 为“全域”, μP_2 为“地面附近”, μP_3 为“高空”的隶属函数。还有 $f_i (i=1, 2, 3)$ 为与 P_i 对应的操作模型。 r 为目标值 本例中 取 $r=0$ 。 $C_{ui} (i=1, 2, 3)$ 表示操作模型的关联度。

式(5.31)~式(5.33)中, 考虑到对模糊规则的学习性, 使用了关联度 C_{ui} 。关联度表示现在的 f_i 和理想的 f_i^* 的关联 越理想 就越接近 1。若关联度 C_{ui} 变高则其状况中操作模型的贡献程度就增加。各操作模型的贡献度 γ_i 通过模糊规则的 if-then 关系和 C_{ui} 的联想推理

得到。

贡献度 γ_i 是把联想存储的推理值 a 进行正规化。模糊控制器的操作量 u^* ，用贡献度 γ_i 按下式合成：

$$u^* = \sum_{i=1}^3 \gamma_i \cdot f_i(y, r) \quad (5.34)$$

式中，作为与操作有关的初始知识，由于没有得到低空操作模型 f_2 和高空操作模型 f_3 ，把全域操作模型 f_1 假设为 f_2, f_3 。因此关联度 C_{u2}, C_{u3} 是比较低的值。联想存储系统中，可以把这样的不完全性知识，用关联度来表示知识的轮廓，用联想的模糊推理和学习来进行知识的简练。

如图 5.15 左上部的贡献度曲线所示，学习前的初始状态，从低空到高空全域上用保守的操作模型 f_1 。然后，用应答信息和操作者的修正信息，通过学习对与各状况对应的操作模型 f_2, f_3 ，进行精练。如图 5.15 右上部的贡献度曲线所示，学习后，对于特征状况 P_2 (低空) P_3 (高空) 向简化后操作模型 f_2, f_3 的操作转移。

(6)稳定性的讨论和操作模型 f_1 的设计 操作模型 f_1 是对直升机参数变动的全域模型 P_1 有效的控制器。如图 5.11 中说明的那样，必须考虑对从低空模型 P_2 到高空模型 P_3 的变动域的同时稳定化补偿器(参见附录)。

操纵者使用了对全域的操作中机体的变位 y 和机体的速度。这些变位和速度(变位的微分)的反馈增益对为 (h_{10}, h_{11}) 。 f_1 的闭环传递函数为

$$P_v = \frac{k_v e_0}{s^4 + k_v d_3 s^3 + k_v d_2 s^2 + k_v e_0 h_{11} s + k_v e_0 h_{10}} \quad (5.35)$$

操作者的变位，根据与速度的比率有关的操作数据，求反馈增益 $(h_{10}=0.189, h_{11}=0.170)$ 。在这个例子中，由于只有一个变动参数 k_v 并且操作模型的结构明确，由式(5.35)的特性方程，应用拉奥非尔 芘茨的稳定条件，得到用于同时稳定的条件

$$k_v > \frac{e_0 h_{11}}{d_2 d_3} k_v > \frac{e_0 h_{11}^2}{d_3 (d_2 h_{11} - d_3 h_{10})} \quad (5.36)$$

式中的各系数为表 5.2, 满足式 (5.36)。由操纵者提取的操作模型 f_1 就成为从低空到高空全域的同时稳定化补偿器。

(7) 模糊规则的设计如 5.3.4 节中说明的那样, 用模糊规则中表示的关系, 利用 BAM 记住模糊规则。但是, 由于是使用关联度 C_{ui} 的知识表示, 要获得 M_{ν} 就要对记住模式乘以 C_{ui} 。

(8) 学习性的讨论 (操作模型 f_i 的简练) 如图 5.15 所示 把来自操纵者指示的修正信号 u 用于教师信号, 对各状况的操作模型进行学习。这时, 如图 5.14 所示, 把从操纵者的操纵杆来的修正信号用作教师信号, 采用有教师学习法。对低空, 高空等语言性表示的各状况, 分别应用求平方误差为最小的 LMS 算法。

各操作模型的控制器, 作为初始知识, 用与从操纵者提取的操作模型 f_1 相同的结构。用 f_1 的参数值 (h_{10}, h_{11}) 作为控制参数的初始值。这里 h_{10} 是对这个控制系统变位 y 的反馈增益 h_{11} 是对变位微分值的反馈增益。

把操纵者的修正信号设为 Δu 把对操作模型 i 的变位 y 的反馈增益的修正量设为 Δh_{10} , 对变位的微分值的增益的修正量设为 h_{11} , 通过学习求反馈增益的修正量就成为把式 (5.37) 的误差 E 最小化。

$$E = \int_1^T (\Delta u - \Delta h_{10} y - \Delta h_{11} \dot{y})^2 dt \rightarrow \min \quad (5.37)$$

在离散时间表示的情况下, 有

$$E = \sum_{t=1}^T (\Delta u_t - \Delta h_{10} y_t - \Delta h_{11} \dot{y}_t)^2 \rightarrow \min \quad (5.38)$$

式中的下标 t 表示时刻, 由式 (5.38) 得

$$\frac{\partial E}{\partial \Delta h_{10}} = 2 \sum_{t=1}^T (\Delta u_t - \Delta h_{10} y_t - \Delta h_{11} \dot{y}_t) (-y_t) = 0 \quad (5.39)$$

考虑到 $|\dot{y}_t|$ 与 $|y_t|$ 相比充分小, 若对式 (5.39) 进行变形, 得到下面的近似公式:

$$\Delta h_{i0} \cong \sum_{t=1}^T \Delta u_t y_t \left/ \sum_{t=1}^T y_t^2 \right. \quad (5.40)$$

同样，进行适当的近似得到

$$\Delta h_{i1} \cong \sum_{t=1}^T \Delta u_t \dot{y}_t \left/ \sum_{t=1}^T \dot{y}_t^2 \right. \quad (5.41)$$

如图 5.15 所示，与随被控对象变动的各状况对应的操作模型的参数，分别用式 (5.40) 式 (5.41) 进行修正 (图 5.15 中的学习 A)。

用基于学习 A 的控制网络的学习，操作模型被简化，随着向理想值趋近 其关联度 C_w 趋近于 1 (图 5.15 中的学习 B)。

作为这个联想网络中学习 B 的效果，如图 5.15 上部所示，学习后，对操纵的贡献度 γ_i 从全域的操作 f_1 向高度的飞行状况的 (低空用的操作) f_2 和 高空用的操作) f_3 转移。这个转移，是使用与联想存储神经网络中的 Hebb 学习规则相同的想法 (按简化操作模型的程度增强关联度)，使各操作模型的关联度 C_w 与其学习进度相对应来变化，很有效。根据关联度的变化，学习后，与各状况对应的操作模型的贡献度 γ_i 变大。

联想推理，与状况相适合，并且对被简练的操作模型进行回想。本实验的例子中，对应学习进度，是按在学习初期 $C_{u1} : C_{u2} : C_{u3} = 7 : 1 : 1$ 学习中期 $5 : 4 : 4$ 学习后期 $1 : 9 : 9$ 变化的。

(9) 模型直升机实验中飞行性能的提高模型悬停实验的学习前和学习后的结果表示在图 5.18 中。横轴为时间，纵轴为传感器输出。实线表示前后，虚线表示左右。

图中表示来自高空的直升机的轨迹。由于学习，悬停性能提高了，学习后变位积分的平均值改善了大约 8%。操作模型各参数为 $h_{2i} < h_{1i} < h_{3i}$ 。而且，全部控制器满足式 (5.36) 的同时稳定化条件。虽然低空中的干扰有增加的趋势，但是通过学习得到的低空用的控制器 f_2 显示出抗干扰的性质。

还有，可以看出高空的循环增益很低，但操作模型有对其补偿的

作用。

本实验，通过学习可以获得与语言表示的各状况的特征对应的操作模型。通过应用模糊规则，可以对基于语言表示的各个特征性参照模型分别进行学习，在谋求学习的快速化的同时，提取获得的知识变得容易了。

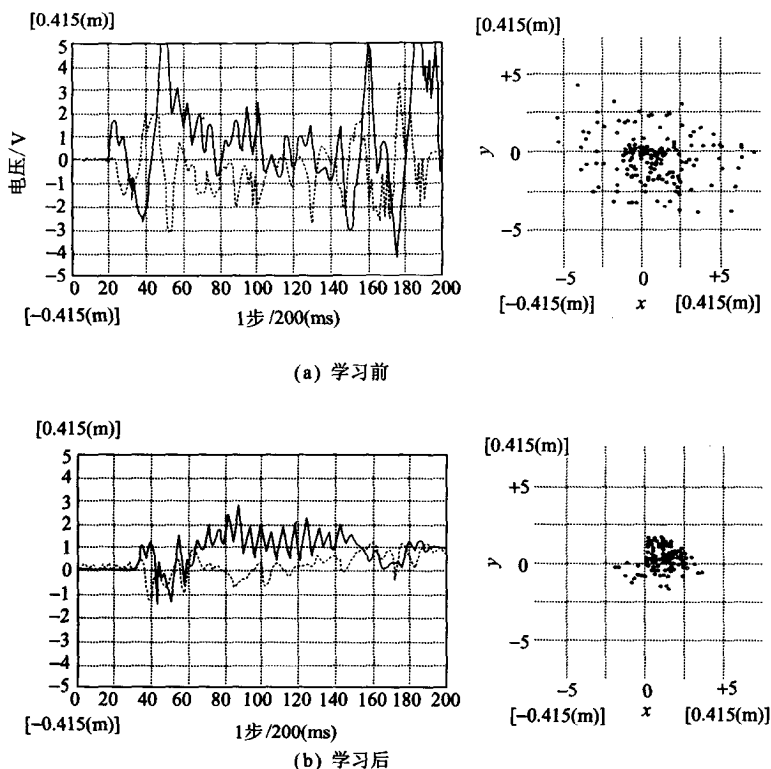


图 5.18 模型悬停实验中飞行数据的推移和来自上方的飞行轨迹

第6章

神经网络和模糊 信号处理应用

- 6.1 神经网络的应用
- 6.2 模糊信号处理的应用

CONTENTS

神经网络和模糊信号处理可以把现实世界的数值信息和模式变换为人类容易理解的方式。如果把这样的特征充分利用，神经网络和模糊信号处理将会在各个领域得到应用。本章将叙述神经网络在模式识别、气象预测以及感性信息处理等方面的应用，还介绍模糊信号处理在建筑相关领域，交通系统，辅助设计和汽车等方面的应用。

6.1 神经网络的应用

对于层次型神经网络，通过增加中间层神经元数，就能够提高任意连续函数的近似精度^[1]。而且我们还知道，通过模糊信号处理，能够对任意连续函数进行近似。这些都是在进行有关神经网络和模糊信号处理时应该想到的重要性质。

6.1.1 应用于模式识别

层次型神经网络在模式识别方面的应用很广泛，如果能成为更实用的系统，那么处理的模式识别种类将更多。若要用整个一个大神经网络来处理这些模式识别，网络规模将变得非常庞大。基于以下理由，把数个小型网络进行综合的神经网络开始受到人们的关注。

学习速度快：由于大规模网络的权值数很多，学习困难，而小型网络学习容易，一般来说收敛速度快；

网络解释容易：网络规模越大，中间层的解释变得越困难。把数个小型网络综合，各网络的解释变得容易，全体的动作也变得容易理解；

中间层神经元数容易确定：小网络的最优中间层神经元数容易确定；

集成化容易：用小型网络，由于相当于 1 个神经元的权值数非

常少，配线部分减少，容易集成化。

这里，将介绍用多个神经网络对重复进行集团化模式进行学习，识别时对各神经网络的输出进行综合判断，求解最终答案的神经网络。

图 6.1 是类组合的神经网络的结构。

由于可以将识别算法分成学习过程和识别过程两部分，所以分别予以说明。

(1) 学习过程综合型网络中，为了减小各个网络（子网络）的学习负荷，把一个网络要处理的类数减少。这时，如何把学习用的各模式分配给各子网络就成为了要点。

文献[3]中，使用了保证至少到第 N 个相近的类的对属于哪个子网络的聚类。开始先求各类的标准模式，然后求各标准模式的欧几里德距离。比如，现在让我们考虑一下 R 类别的文字识别。由于类别数为 R 开始要准备 R 个子网络。子网络 i 中从类 i 开始按相近的顺序直到第 N 个类成为识别对象。这里，把子网络 i 的类称为主类，而把相近的到第 N 个的类称为子类。

在构成某个子网络的过程中，如果其主类和子类已经和前面的子网络一样的话，就将这个子类消除。如果子类全部被消除，就消除其子网络。

用这样的方法，就可以保证类别数以下的子网络中至少到第 N 个相近的类的对属于哪个子网络。

还有，对各子网络的规模，希望它们相同的时候很多。这种情况下，就进行涉及类数少的子网络的结合。

以下把学习过程作为算法进行了归纳：

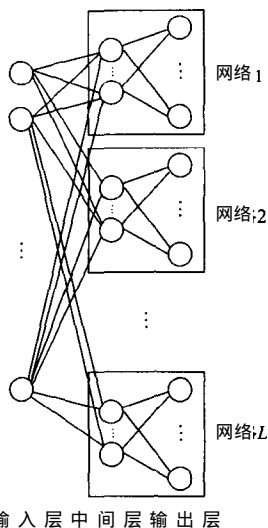


图 6.1 类组合的神经网络结构

第 1 步制作各个类的标准模式；

第 2 步 设置 $i=1$ ；

第 3 步 找 N 个从 $P(i)$ (标准模式 i) 开始到第 N 个相近的标准模式 $P(j)(j \neq i)$ ；

第 4 步 如果 $P(i)$ 和 $P(j)$ 已经属于相同的类，消除 $P(j)$ ；

第 5 步 如果 $P(j)$ 为 0 个，不做子网络，否则，构成 $P(i)$ 和 $P(j)$ 的 $(N+1)$ 个类所属的子网络；

第 6 步 如果 $i < R$ (R 为类别数)，则 $i=i+1$ 返回到第 3 步；

第 7 步 设置 $k=1$ ；

第 8 步 设置 $n=k+1$ ；

第 9 步 考虑两个子网络，如果它们处理的类别数在 $N+1$ 以下，则把它们进行结合；

第 10 步 如果 $n < M$ (M 为第 6 步结束时的子网络数) 则 $n=n+1$ ，返回 Step9；

第 11 步 如果 $k < M$ 则 $k=k+1$ 返回到第 8 步。

各子网络用反向传播算法进行学习。

(2) 识别过程 识别过程中，要根据各子网络的输出进行综合判断。首先叙述算法然后举例说明。

识别过程算法步骤：

第 12 步 向网络的输入层输入模式；

第 13 步 对每个网络独立计算输出；

第 14 步 在全部子网络的答案中，把在其他子网络中被否定的答案从候选中取消；

第 15 步 在最终候选中，把与相对应类的平均向量的距离为最小者作为答案。

上述方法中，样本空间中关于相近的类别，使其属于同一子网络，进行反向传播的非线性识别。

关于较远的类别是基于这样的原则，即使用与子网络学习的模式

的平均向量的距离，用线性方法进行识别。

例 6.1 文献[3]中 作为识别实验 应用本方法对 20 人的面部图像进行了识别。这里用具体例子详细说明其算法。

首先是学习过程。由于对象人数为 20 类别数也为 20。首先，考虑识别各个类的共计 20 个子网络 分析一下到第 5 个(取 $N=5$) 相近的类(第 3 步)。将这些在表 6.1 中列出。

然后 执行第 4 步、第 5 步。结果表示在表 6.2。

注意表 6.1 中的子网络 3 类 3 和类 12,11,14 这组 由于在子网络 1 中已经被处理，所以被消除。同样，类别 3 和类别 10 及 20 这组，在子网络 2 中已经被处理，所以被消除。由于子网络 3 中应该处理的类的组在子网络 1 和 2 中全部已经被处理过，所以子网络 3 被除去。

表 6.1 接近第 5 个的类

网络 1	1,12,3,11,14,9
网络 2	2,12,10,3,20,11
网络 3	3,12,14,20,10,11
网络 4	4,12,14,13,3,11
网络 5	5,12,16,9,4,11
网络 6	6,18,9,12,8,20
网络 7	7,10,17,20,2,3
网络 8	8,12,6,9,3,20
网络 9	9,6,12,1,8,19
网络 10	10,20,3,18,12,2
网络 11	11,12,3,1,18,14
网络 12	12,3,11,1,4,8
网络 13	13,4,12,3,8,6
网络 14	14,3,12,20,4,11
网络 15	15,6,8,9,18,20
网络 16	16,5,12,6,9,11
网络 17	17,7,10,15,20,2
网络 18	18,20,6,11,3,12
网络 19	19,9,6,1,18,12
网络 20	20,18,10,3,14,12

表 6.2 聚类结果

网络 1	1,12,3,11,14,9
网络 2	2,12,10,3,20,11
网络 3	4,12,14,13,3,11
网络 4	5,12,16,9,4,11
网络 5	6,18,9,12,8,20
网络 6	7,10,17,20,2,3
网络 7	8,3
网络 8	9,19
网络 9	10,18
网络 10	11,18
网络 11	13,8,6
网络 12	14,20
网络 13	15,6,8,9,18,20
网络 14	16,6
网络 15	17,15
网络 16	19,18

表 6.3 为用第 9 步进行子网络结合的结果。比如，表中子网络 7 是表 6.2 的子网络 7、8、9 被结合后的网络。这样，最终就构成了 11 个子网络。

识别过程中，首先要向各子网络的输入层输入未知模式，然后分别独立计算输出。

下面用例子说明识别过程中的判断。表 6.4 中表示了输入属于类 16 的模式时各子网络的答案(黑体字)。这里，尽管使用子网络 4 进行了学习，但候选 11 和候选 12 却没有成为子网络 4 的答案。因此 候选 11 和候选 12 就在子网络 4 中被否定(第 14 步)。

表 6.3 最终聚类结果

网络 1	1,12,3,11,14,9
网络 2	2,12,10,3,20,11
网络 3	4,12,14,13,3,11
网络 4	5,12,16,9,4,11
网络 5	6,18,9,12,8,20
网络 6	7,10,17,20,2,3
网络 7	8,9,19,10,3,18
网络 8	11,13,8,6,18,16
网络 9	14,17,15,20
网络 10	15,6,8,9,18,20
网络 11	19,6,1,18,12

表 6.4 各子网络的输出

网络 1	1,12,3, 11 ,14,9
网络 2	2, 12 ,10,3,20,11
网络 3	4, 12 ,14,13,3,11
网络 4	5,12, 16 ,9,4,11
网络 5	6,18,9, 12 ,8,20
网络 6	7,10,17,20,2, 3
网络 7	8 ,9,19,10,3,18
网络 8	11,13,8,6,18, 16
网络 9	14,17, 15 ,20
网络 10	15 ,6,8,9,18,20
网络 11	19,6,1,18, 12

同样 候选 3 候选 8 也分别在子网络 1 和 5 中被否定，这时候剩下候选 15 和 16。最后，根据第 4 步，在剩下的候选中，计算与对应类平均向量的距离，取最小者为答案。

这时输入向量，由于与类 15 的平均向量相比，更接近类 16 的平均向量，所以最终的识别结果为类 16。

在改变第 3 步的 N 的情况下，对识别率的影响如图 6.2 和图 6.3 所示。图 6.2 为对学习数据的识别率，图 6.3 为对未学习数据的识别率。

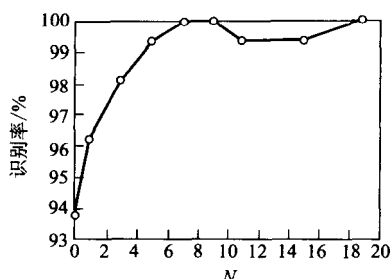


图 6.2 对学习数据的识别率

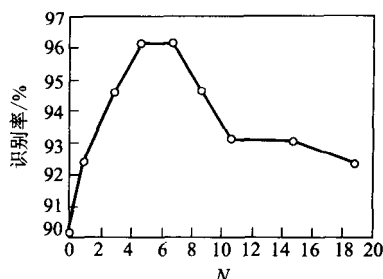


图 6.3 对未知数据的识别率

注 6.1 一般，模式识别中所用数据的前处理非常重要。关于面部图像，在本系列书中的《语音与图像的数字信号处理》一书中有详细叙述，请参照。

注 6.2 在把数个小型网络综合的网络中，除了文献[3]的网络以外，还有 CombNET-II^[4]，M. Jordan 等人的 HME (hierarchical mixtures of experts)^[5]，及下节要介绍的 Glove-Talk 等形式。与 CombNET-II 的主要区别在于类簇重复进行。因此，当模式有较大变动时，前面介绍的方法可以得到较高的识别率^[6]。还有，学习次数少 是 HME 的显著特征。

可以看出，对于学习数据，随着 N 的增大识别率也提高，而对于未学习数据 N 变得再大，识别率都会降低。这是由于网络的规模变大，容易陷入过学习的原因。

6.1.2 应用于手语-声音接口

给身体有残疾的人提供辅助机器的需求，随着老龄化社会的发展，今后会越来越高。这里，以手语向声音变换的接口中的应用为例的 Glove-Talk^[6] 予以说明。

神经网络擅长于学习复杂的输入输出关系，G. Hinton 等人注意到了这个问题，在手的动作和声音的对应关系学习中应用了神经网络

络。在这里，也将数个小型网络进行了有效综合。

图 6.4 中表示了手语-声音变换的 Glove-Talk 的概念图。在输入部，使用了在人工真实感（virtual reality）中经常应用的 Data-Glove（VPL 公司制作）Data-Glove 是配带有传感器的手套，在各手指上装上有两个角度的传感器。

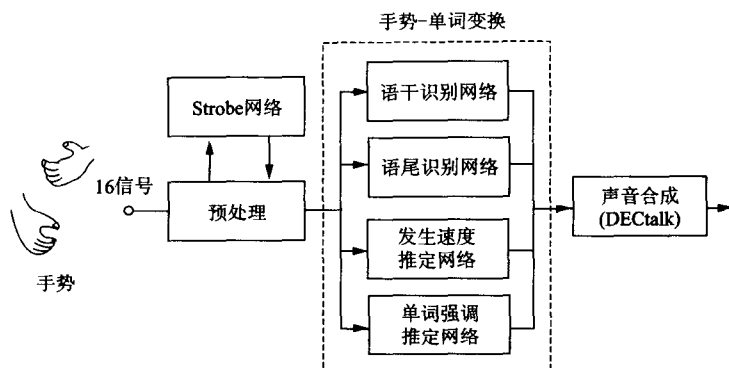


图 6.4 进行手语-声音变化的 Glove-Talk 概念图

由这些传感器对距某个固定点的 x, y, z 的变位 胳膊的横摇 纵摇 偏摇 共计 16 个数据进行计测。采样频率为 60Hz。在声音合成部使用了 DEC 公司的 DECtalk。

在 Glove-Talk 中使用的神经网络，有对手语部分的检出网络和对输入数据的声音合成部进行控制的四种网络，即单词的语干识别，单词的语尾（变形）识别，发声速度的推定和单词的强调推定等网络。下面分别予以说明。

(1) strobe 网络 Glove-Talk 先将 Data-Glove 的信号存储在前处理部的暂存器中。判别是否发出手语，即手语部分的检出，在手语-声音变换中很重要，必须在一开始进行。完成这一任务的是 strobe 网络。对 strobe 网络输入以下 5 种信号成分：

①～③ 手的位置变化成分 $\Delta x, \Delta y, \Delta z$;

④ 手的运动速度 (绝对值) $\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$;

手的运动加速度。

将各成分的 10 个样本, 换算成时间以 167ms(1/60×10) 输入给网络。所以输入层神经元数是 50。

这些输入信号, 经过中间层(10 个神经元)进入输出层。输出层为 1 个神经元, 判断目前样本是否为手语。

(2) 语干识别网络 Glove-Talk 中, 以 66 个基本单词为识别对象。各个单词最多有 6 种语尾, 具体为 -s(复数形式)、-ed(过去式)、-ing(进行时)、-er, -ly, 无变化。通过把这些进行组合可以处理 203 个单词。

对语干识别网络, 输入来自各个手指的角度传感器的输出(10 个), 胳膊的横摇, 纵摇, 偏摇的各自的正弦成分和余弦成分, 合计 16 个数据。之所以用正余弦, 是因为, 直接用角度的话会产生不连续的问题, 而用正余弦则可以有效地避免。

网络中的输入层、中间层和输出层分别用 16, 80 和 66 个神经元。输出层的 66 个神经元的输出按以下的方法正规化为 1。

$$y_i = \frac{\exp(net_i)}{\sum_{j=1}^{66} \exp(net_j)} \quad (6.1)$$

式中, y_i, net_i 分别表示神经元 i 的输出, 神经元 i 的输入和。

因此, 输出层的各神经元输出就表示分别与之对应的单词的概率。

(3) 语尾识别网络语尾是根据胳膊的单一运动, 比如用胳膊向上的动作或向下的动作来表示。对语尾识别网络的输入为 $\Delta x, \Delta y, \Delta z$ 三种, 各 10 个样本全部为 30 个, 中间层和输出层中各有 10 个和 6 个神经元。输出层神经元的输出中, 与语干识别网络的情况相同, 输出和被正规化为 1。

(4) 发声速度推定网络对发声速度推定网络, 输入手的运动速

度（绝对值）和手的运动的加速度两种，输入 20 个样本。中间层神经元数为 15，输出层神经元数为 8。输出层的 8 个神经元分别输出从 0 到 1 的值，并根据它们的组合，能够改变每分钟从 130 到 340 个单词的发声速度。

(5) 单词强调推定网络对单词强调推定网络的输入，与发声速度推定网络的情况相同。根据手动动作的大小，推定单词的强调情况。输出层神经元为 1 个，使用 S 函数取阈值为 0.5 来决定是否对单词进行强调。

这样，用动作的大小和速度可以控制合成声音的强弱和速度。对 203 种手语单词进行实验的结果，得到了 99% 以上的识别率。

这种应用不仅局限于 Glove-Talk 中用的神经网络，由于神经网络具有很高的学习能力，能够与用户的试验相对应进行网络的学习，因此，容易构成与用户适合的系统。想与其他用户对应时，可以制作新数据让其再学习。还有，实际应用时的计算量，与学习时相比要少得多。

6.1.3 应用于气象预测

如果将神经网络的输入输出层二维配置的话，就可以容易地处理图像。这里介绍从气象雷达得到的云的图像（参见 6.3 节）进行降雨，降雪预测，大大提高预测精度的例子^[7]。

由于三维空间的大气状态和地形及由它们的相互作用而气象变化复杂，要正确预测天气是非常困难的。

过去曾有几种气象预测的方法，其代表是基于图像的平行移动的线性预测法。这是把地形效果用数学公式详细描述后，假定云的形状不变化，通过计算云的移动速度，方向来进行经过一定时间后的降雨，降雪预测。然而，实际上，云的形状时时刻刻都在变化，这种方法的前提本身就存在问题。

这里介绍的方法，把从气象雷达得到的图像原样输入给神经网络。因此就成为把云作为点的集合来处理，可以毫无疑问地对应云的

形状变化。

图 6.5 表示的是应用神经网络的降雨降雪预测系统的概要。气象雷达图像在 240km 四方的领域中以每 1km 四方进行区分,生成 57 600($=240 \times 240$) 个降雨降雪量数据。本系统基本上是预测 5 分钟后的降雨降雪,对某地点的预测用近旁 6km 四方内的数据。

具体地 见图 6.5 上部所示,在 6km 四方的范围内的斜正方形领域的 25 个地点的降雨降雪量数据作为神经网络的输入,经过中间层(神经元数 3 个),输出预测量。预测这样的 1 个地点的降雨降雪量的小型子网络,分布在 240km 四方范围内,也就是说有 57 600 个。

将时间间隔 5 分钟拍摄的两张雷达图像作为输入输出数据对,把它们准备成几十组进行网络的学习。进行 5 分钟以上时刻预测的情况下,如图 6.5 所示,可以把网络输出只以必要的次数重复向输入进行反馈。

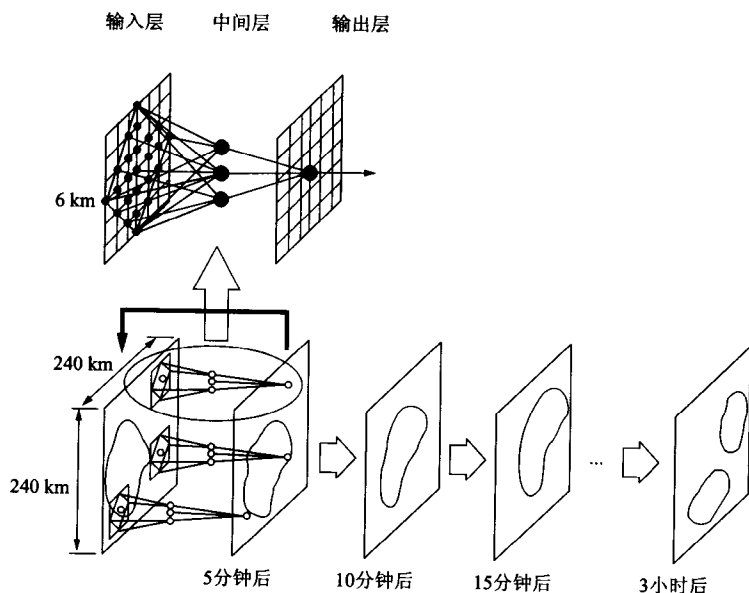


图 6.5 使用神经网络的降雨降雪预测系统概要

本系统的一个神经网络中有 $82(=25 \times 3 + 3 \times 1 + 3 + 1)$ 个(参见 6.4 节)权值 这是因为有 57 600 个网络, 必须缩短学习时间。且提高预测精度。

为了缩短学习时间, 应用了可以检出权值振动现象尽快进行控制的 Kick Out 法^[8]。根据文献[5]可知, 用该方法, 学习时间大约减少 1/10。为了提高预测精度, 防止过学习, 不使权值过大, 用了增加限制的方法。这样, 预测误差能减少约 1/5~1/10。

图 6.6 表示的是实际云层的活动和其预测。图(a)是实际 3 小时后的图像 图(b)是用本系统预测的 3 小时后的图像, 图(c)是用基于线性预测的方法得到的图像。与以前的方法比, 可以看出基于神经网络的本系统预测精度比较高。

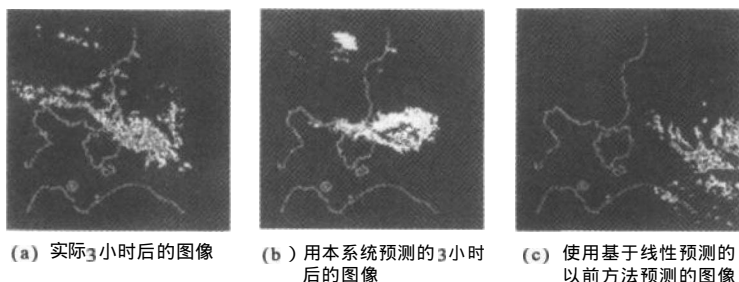


图 6.6 实际云的变动及其预测图像 札幌综合信息中心提供)

注 6.3 气象雷达实际拍摄的有雨和雪, 但是这里为了简单起见 全部表现为“云”。

注 6.4 考虑到中间层, 输出层中有偏移项, 对全体个数增加 3 和 1

6.1.4 电子宠物中的应用

近年来, 把人的感性在工学上进行处理的感性工学受到了关注。

www.aibbt.com 让未来触手可及

由于感性的数学处理很困难，因此可以说是适合于神经网络和模糊信号处理的领域。

作为神经网络的一个独特应用是“尤罗娃娃”^[9]。它是以与在越来越复杂化多样化的社会中生活的现代人共存，并且可以柔性交流的电子宠物为目标的。尤罗娃娃具有判断从人的声音的抑扬顿挫到喜怒哀乐的感情功能。

它可以在欢乐时笑，悲伤时哭，不高兴时生气，厌倦时打哈欠。对于其表示，由于应用了计算机图形学技术，可以由软件来增加面部表情的种类。

尤罗娃娃的性格可以根据神经网络的学习数据变化。人很自然想用语言进行交流，在公开表演中，很多男女老少都跟不能说话的尤罗娃娃说了话。

这里介绍尤罗娃娃的扩展形式“麦克”^[10]

表 6.5 麦克的感情识别

感 情	发 声
喜悦(幸福、满意、高兴、愉快等)	兴奋,气力强,语末上扬
生气(发怒、愤慨、不愉快)	语末的压抑
吃惊(震惊、困惑、惊愕、意外)	尖叫声,兴奋的声音
悲伤(悲叹、悲哀、孤独、空虚感)	弱声,微弱声,空虚的声音
厌恶感	不高兴,厌恶,恶心的声音
开玩笑	快活,不认真的声音
恐惧	胆怯,剧烈,高调的声音

(1) 麦克的设计 麦克能够识别表 6.5 所示的 7 种感情 并分别对其进行反应。比如，在文章的最后发声上扬的话，识别为表示幸福感 满足感 愉快的‘喜悦’感情并进行反应。

图 6.7 表示的是麦克的处理流程图。下面对各个部分予以简单说明。

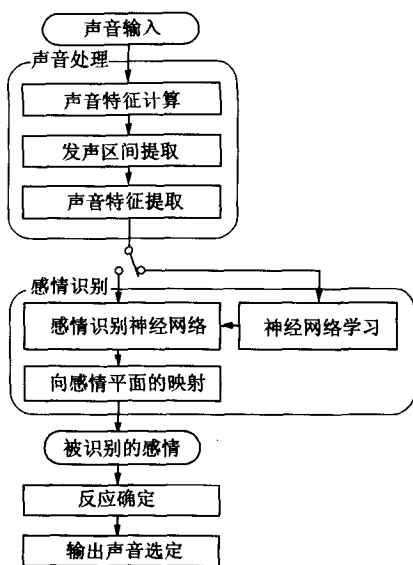


图 6.7 麦克的处理流程

声音特征的算出：声音中有意思内容信息和音律信息。意思内容信息是谱信息，音律信息是声调信息。以前的声音识别中音律信息不是必要的，而在感情识别中两者都是必要的。选择与意思内容对应的特征量 LPC (Linear Predictive Coding)参数，对应于音律的特征量 着眼于声音振幅 时间变化 及音调 由 LPC 分析算出使用。

发声区间的提取：根据声音功率的大小，提取发声区间。

声音特征的提取：归纳 10 帧声音特征，作为向感情识别部分的输入。

感情识别神经网络：为了识别表 6.5 表示的 7 种‘感情’和‘平静’共计 8 种 使用了 8 个子网络。就感情来说，感情间的识别非常困难，因此，使用了对各个感情识别特殊化的子网络。各子网络中，输入层神经元有 150 个，输出层神经元有 1 个，中间层神经元数从 20 个到 30 个不等。

向感情平面的映射：根据各子网络的输出最终判断感情，向图 6.8 所示的感情平面映射。另外，指定 x 轴为愉快(+) 不愉快(-)， y 轴为强感情(+) 弱感情(-)。

⑥ 反应确定：驱动计算机图形，在显示器上表示图 6.8 那样的反应。

⑦ 输出声音的选定：同样，进行声音的反应。

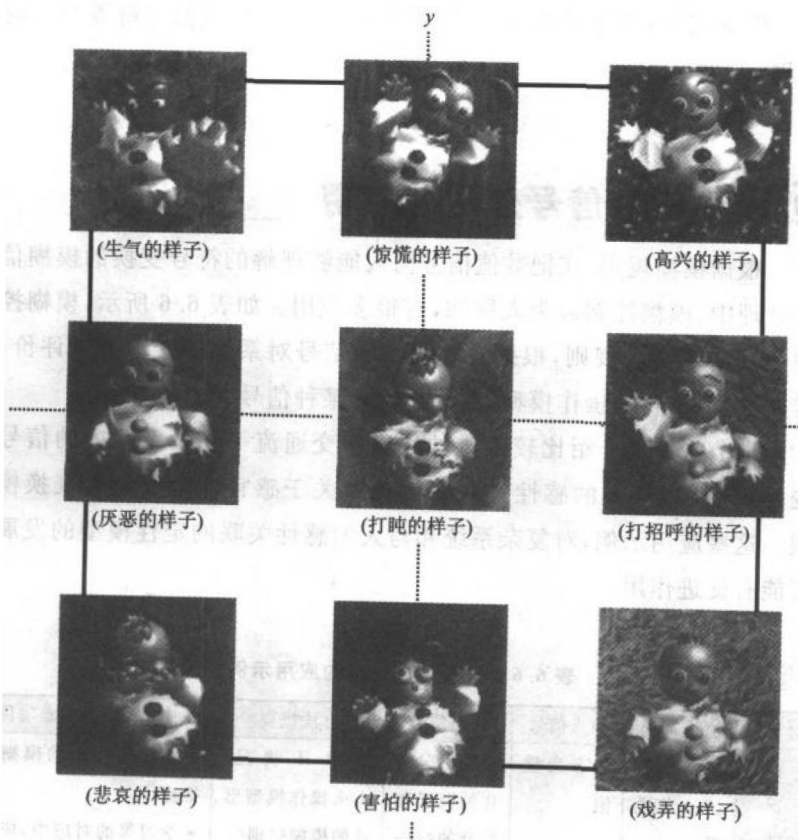


图 6.8 MIC 的感情表现

(2) 尤罗娃娃和麦克的扩展对这样的尤罗娃娃和麦克，考虑了各种各样的扩展。由于使用了人工现实感而带有触觉功能，就增加了握手的效果。也由于加入根据声音的感情判断，根据握手的强度能够判断对应的方法。

还有 由于没有进行声音识别 就不依存于语言。因此 在日本和美国有各种尤罗娃娃，也有用计算机网络进行感性交流的实验。不仅根据声音，还有根据音乐识别感情来进行反应的 MUSE 也正在开发中^[10]

除此之外，引进视觉，嗅觉等感觉也很有可能，我们期待着今后的发展。

6.2 模糊信号处理的应用

根据模糊规则，在把数值信号向人能够理解的符号变换的模糊信号处理中，模糊控制最为人所知，有很多应用。如表 6.6 所示 模糊控制是基于定性的规则，根据系统的状态信号对系统的变动进行评价，转换操作量、转换操作模型（控制器），是某种信号变换的模型。

本节将简要介绍比较困难的地震和交通流等复杂系统中的信号变换模型，和与人的感性关联的设计及关于感官评价的信号变换模型。这些应用示例，对复杂系统和与人的感性关联的定性模型的发展可能有促进作用。

表 6.6 模糊信号处理的应用示例

对 象	应用举例	输入信号	输出信号	方法概要	本书相关联的基础项目
状况变化模型	模糊控制 (5.4 节说明)	状态变量值	对应状况变化的模型和模型的操作量	使用 if 状况, then 操作模型形式的模糊规则	<ul style="list-style-type: none"> 应用第 5 章的模糊推理 学习等的对应中, 应用第 5 章的模糊神经网络

续表 6.6

对 象	应用举例	输入信号	输出信号	方法概要	本书相关联的基础项目
复杂系统模型	地震主要震 动预测模型(6.2.1 节说明)	初期微动 能谱和震 动方向数 据值	主要震动的 大小预测	考虑由于例外规 则表示的结构化 神经网络	<ul style="list-style-type: none"> • 某种模糊神经网络中接近图 5.4(c) 的类型 • 学习中,应用第 2 章的反向传播
	电梯群管 理系统中 等待时间 的预测模 型(6.2.2 节说明)	来自各层 的大厅呼 叫发生数 据和各电 梯内的到 达点呼叫 发生数据	对应交通流 状况的预测 模型的等待 时间分布	使用 if 交通流的 状况, then 对应 其状况的等待时 间预测模型形式 的模糊规则	同上
感性模型	汽车设计 评价模型 (6.2.3 节 说明)	汽车各部 分形状数 据	对应其形状 的印象用语	根据模糊关系表 示输入→输出 而且根据逆推理 实现输出→输入	<ul style="list-style-type: none"> • 对本书关联项目中实现相同的推理,逆推理,第 5 章的模糊联想推理有效,应用输入→输出,输出→输入双向推理
	驾驶员操 作的官能 评价模型 (6.2.4 节 说明)	车的视觉 变化,加 速器操作 量	操作的感官 评价价值	根据 ARMA 模 型用模糊规则实 现,由模糊推理 输出评价价值	<ul style="list-style-type: none"> • 应用第 5 章的模糊推理(实现推理法的扩展) • 实现与模糊联想推理中的模糊性控制相近的推理

还有一些方法本书没有涉及到,通过增加其他方法代用的可能性说明,来显示技术领域的关联性。基于模糊示例^[11]思想的信号处理,是这些方法中相同的概念。

6.2.1 建筑相关领域中的应用

我们以复杂的地震主要震动预测模型进行说明。该预测模型是根据地震初期微动的数据预测主要震动的大小，利用从初期微动到达主要震动的时间差，能够进行建筑设施的防护，即居住者避难引导的自动表示，电梯和煤气的自动关闭。图 6.9 所示是主要震动的大小预测模型。

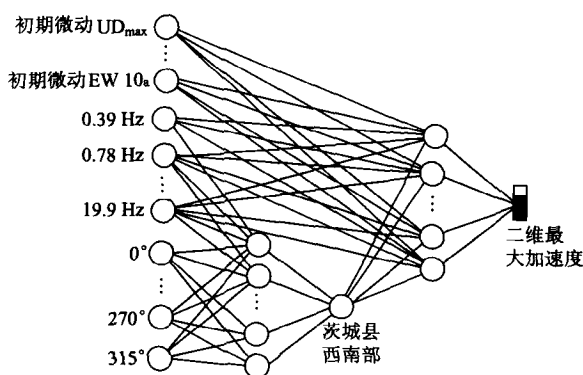


图 6.9 主要震动大小的预测模型

本预测模型是由对一般主要震动的二维最大加速度进行预测输出的主神经网络，和对茨城县西南部的例外知识进行补正的子神经网络，根据表示其关联的规则，构成的构造化神经网络，即由某种模糊神经网络构成。

存储一般知识的主神经网络中，根据表示初期微动全体特征的 4 个输入（图中 初期微动 UD_{max} ~ 初期微动 EW_{10a} 的 4 个输入：上下方向成分的最大值，上下方向成分的最大值的 10 个点的平均值，东西方向成分的最大值，来自东西方向成分最大值的 10 个点的平均值），和表示初期微动能谱的 50 个输入（图中，用 0.39 Hz，

19.9Hz表示的 50 个输入 从 0.39Hz 到 19.9Hz 的能谱的 50个分割) 来输出主要震动的二维最大加速度。

存储例外知识的子神经网络中, 输入把朝东作为 0 度的震动方向数据 (图中 用 $0^{\circ}, \dots, 270^{\circ}, 315^{\circ}$ 表示的 8 个输入 定义成从 0° 开始每隔 45°), 对带来例外结果的茨城县西南部进行输出。这里, 根据 S 函数定义输入的震动方向数据。

各个神经网络应用反向传播来存储实际输入输出数据。而且, 子神经网络, 为了补正作为例外知识的最终输出的二维最大加速度值, 进行再学习以提高预测精度。图 6.10 是主要震动的预测结果和实测值的比较。

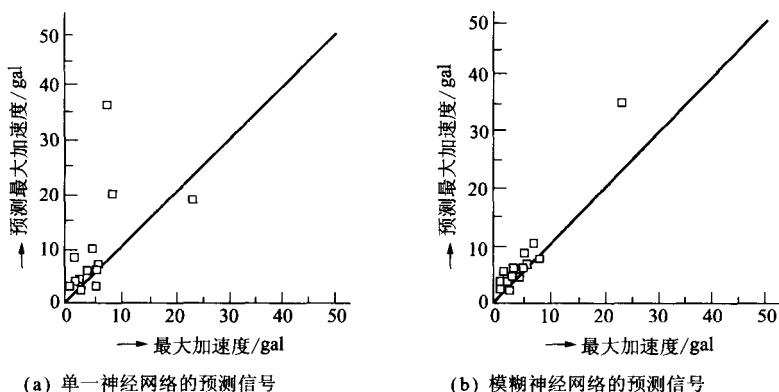


图 6.10 主要震动的预测结果比较

图 6.10(a)表示存储一般知识的主神经网络的预测结果, 图 6.10 (b)表示模糊神经网络的预测结果, 作为子网络把例外知识层次性地附加给了前面的网络。根据例外知识的补正效果, 与图 6.10(a)所示的结果相比 图 6.10(b)中的预测结果平方误差减小了约 45%。以上可见, 根据经验找出为作为例外的知识的情况下, 由层次性网络构成的模糊神经网络是有效的方法。

6.2.2 交通系统领域中的应用

图 6.11 表示的是模糊神经网络在作为纵向交通系统的电梯群管理系统中的应用。

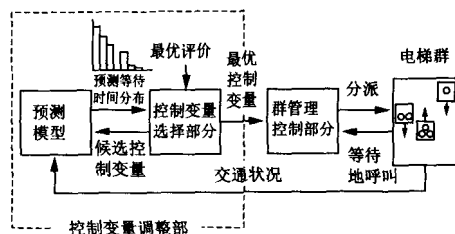


图 6.11 系统构成

大规模高层建筑中，使用对数个电梯进行统一控制的管理系统。群管理系统，对应乘客的乘点呼叫，从电梯群中选择分派服务的电梯。被分派的电梯，使预报指示灯发亮，向在乘点等待的乘客提示。对使用者来说，预报变更就意味着劣质服务。

还有，让乘客长时间等待也不行。群管理系统就是为了不产生预报变更和长时间等待的发生而进行分派控制。

但是，由于事先不知道是否有乘客以及要去的地点等信息，所以存在很多不确定的因素。进行分派的群管理控制部为了能对应诸多交通状况，就要有很多控制变量。控制变量由于对分派有间接的效果，要到达最优分派控制而进行控制变量的调整就很困难。

本系统中，在控制变量调整部的预测模型的构成上应用了模糊神经网络。图 6.12 中构成的预测模型，是交通状况 D (图中 D_1 表示乘客平均发生间隔, D_2 表示来自标准的乘客平均发生间隔, D_3 表示面向标准层的乘客平均发生间隔) 中, 使用控制变量 α 时的输出等待时间分布 y 的模型。

本预测模型是图 5.4(c)的 (前件 + 后件 + 逻辑)型的结构。评价规则前件部的交通状况的隶属函数在图左侧，与规则后件部的状况对应的子预测模型的输入输出函数在图右侧，图中央是推理部。

在规则表示中，应用了高木、营野的模糊模型表示，在规则的后件部不用隶属度函数而用输入输出函数。

图 6.12 左侧的交通状况，根据来自各层的大厅呼叫信号和各电梯内的去向呼叫信号的数据算出，输入给评价交通状况的各隶属函数。

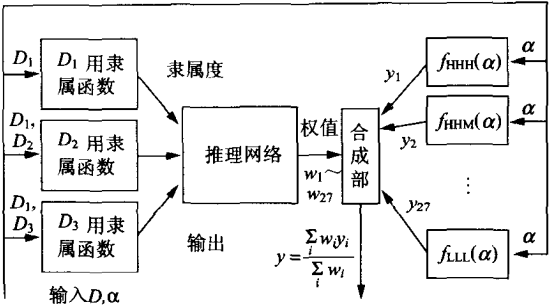


图 6.12 预测模型结构

图 6.12 中右侧的函数 f ，是与各交通状况（出勤时间带，午饭时间带等）对应的子预测模型，各个模型输出在对分派控制中使用控制变量 α 时的等待时间分布。 $f_{HHH}, f_{HHM}, \dots, f_{LLL}$ 的下标 H, M, L 为表示混杂程度的三种交通状况 D_1, D_2, D_3 分别为 high (高), middle (中等), low (低)。

这些模型，用单个神经网络构成，使用反向传播学习法，根据实际输入数据进行在线简练。根据通过学习得到的预测模型，使控制变量的最优选择成为可能。在电梯台数 8 台 电梯速度 420m/min 电梯规定乘员 24 人，服务层段为 15 个层段 (B1, 1~3, 22, 30~39 层) 的条件

下做了详细的仿真实验，进行了控制变量的最优选择。其控制结果表示在图 6.13。

图中，虚线是以前系统的结果，实线是使用本节所介绍的方法的新系统的结果。图 6.13(a)表示对应乘客发生数的平均等待时间，图 6.13(b)表示长时间等待率（等待时间为 60 秒以上的呼叫比率）。根据学习对对象模型进行改善，根据其模型调整控制变量，这样即使乘客发生数增加，也可以抑制平均等待时间或长时间等待。

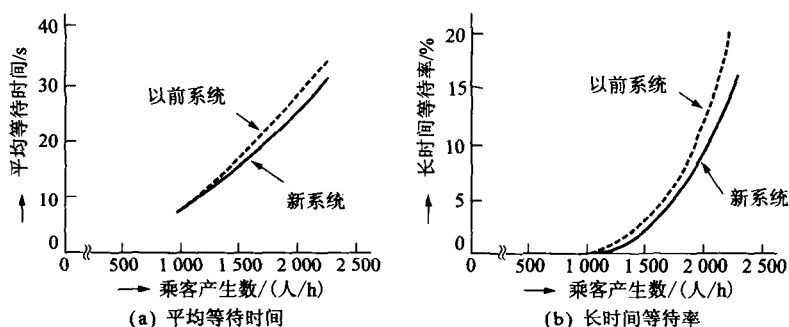


图 6.13 仿真结果

像这个系统，可以说，在粗略的模型结构，即 if 交通流状况，then(对应其状况的等待时间预测模型)的规则由经验得到的情况下，用基于规则的层次结构的模糊神经网络，比一个大规模神经网络，在学习的快速性和人类易于接受的程度上更为有效。

以模糊神经网络为代表的结构化神经网络的构筑工具，在有复杂结构的系统的模型化上很有效。图 6.14 表示的是结构化神经网络的构筑工具。本工具是对前述的电梯系统模型化的模糊神经网络进行通用化得到的。

图 6.14(a)表示系统的构成示例，图 6.14(b)表示作为其构成要素的子系统。可以把子系统看作一个神经网络。学习时，与反向传播

相同，根据来自输出一侧的误差信号 ζ 调整内部参数。通过把这样的子系统构成层次式，更复杂的系统的模型化成为可能。本工具以 Enfant (Enhanced fuzzy and neuro tool) 命名，可以很方便地由因特网免费得到。

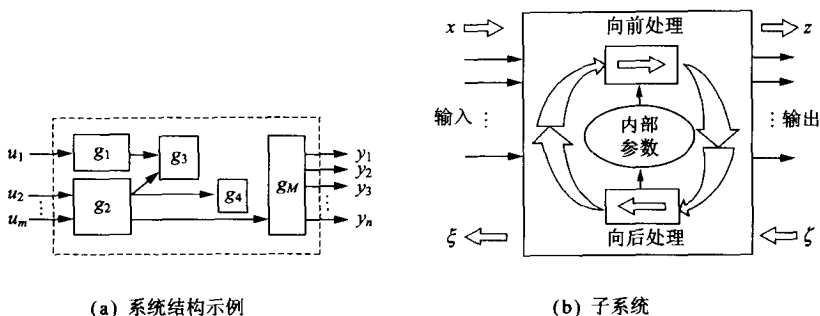


图 6.14 结构化神经网络构筑工具

6.2.3 辅助设计领域中的应用

下面是汽车辅助设计系统的应用举例。本系统可以获得车的各部分形状数据和对其形状的印象用语的关系，实现来自形状的印象推理和得到有印象意思的形状的逆推理等，可以说适用于“鲜明度”等与人的感性相关联的模型。

以表 6.7 的样车 对图 6.15 所示的形状数据进行测量，根据对其形状的问卷调查获得印象用语。

但是要把形状数据，用如图 6.16 所示例子那样的光滑的隶属函数 变换成与小、中、大对应的形状标识 图中 由于是评价车高的隶属函数 所以变换成“车身低的车”；“难说出高低的车”；“车身高车”等标识。

表 6.7 样本车一览表

1. Persona	21. Benz560SEL
2. OPEL Vectra(藏青色)	22. BMW735i
3. Saab9000CD	23. Celsior
4. Legend	24. Fairlady Z
5. Maxima	25. RX-7
6. OPEL Omega	26. Celica
7. Audi80	27. RENAULT Ariven
8. Accord	28. MR-2
9. Primera	29. PORSCHE911
10. Camry	30. PEUGEOT 405 * M116
11. PEUGEOT 405 * SRI	31. Skyline
12. Concerto	31. Mark II
13. Sprinter	33. OPEL Vectra(金色)
14. Sunny	34. Cefiro
15. Ciric	35. Carina ED
16. Gemini	36. BMW535i
17. Corolla	37. Familia
18. Daimeler XJ40	38. Laurel
19. Lincoln continental	39. Galant
20. Infinity Q45	40. Passat GT

根据模糊关系获得形状标识和印象的关系。这一模糊关系用下式表示：

$$A \circ R = B \quad (6.2)$$

式中 A 为表示各形状标识等级的向量， R 为关系矩阵 B 为表示印象用语等级的向量， \circ 表示模糊关系中的合成规则。

这样获得的模糊关系 R 的逆推理（由印象用语 B 向形状标识 A

的推理)的结果表示在表 6.8 表 6.9 中。

表 6.8 表示由实际车的数据产生的结果，表 6.9 表示作为以未知假设数据使用本表下部的印象用语时的推理结果。模糊关系的合成，如表 6.8 和表 6.9 所示，得到了在下限和上限的区域中表示的必然解（交集：任何解的范围都是共同解）和可能解（并集：包含获得解的最小解和最大解）。在式(6.2)中要注意这点，必须与通常的合成规则区别开来。

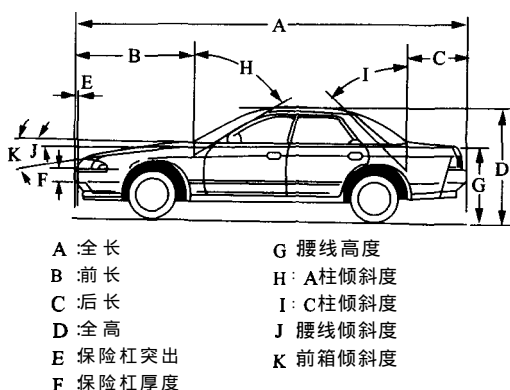


图 6.15 小轿车中各形态要素

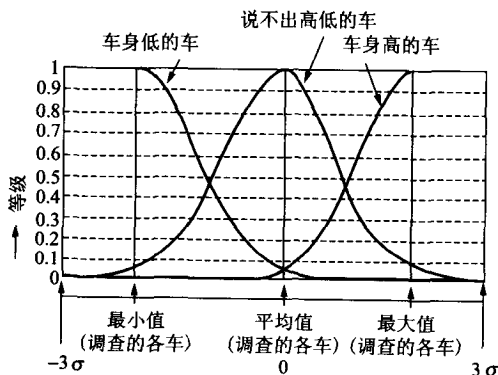


图 6.16 隶属函数的形状

表 6.8 推理解与实际车的值的比较(之一)

形态要素	根据照片测得的实际车的尺寸	推理解		形态要素	根据照片测得的实际车的尺寸	推理解	
		必然解 [下限,上限]	可能解 [下限,上限]			必然解 [下限,上限]	可能解 [下限,上限]
①	0.318	[0.303,0.317]	[0.303,0.318]	①	0.317	[0.308,0.316]	[0.306,0.321]
②	0.282	[0.284,0.301]	[0.283,0.303]	②	0.269	[0.269,0.290]	[0.269,0.303]
③	0.094	[0.061,0.101]	[0.063,0.101]	③	0.127	[0.104,0.134]	[0.088,0.137]
④	29.688	[28.200,30.020]	[27.956,30.052]	④	31.124	[29.173,30.836]	[28.523,31.759]
⑤	24.336	[20.219,26.496]	[19.395,26.496]	⑤	27.109	[20.720,26.589]	[17.234,27.333]
⑥	0.040	[0.043,0.071]	[0.041,0.072]	⑥	0.065	[0.055,0.080]	[0.049,0.094]
⑦	0.309	[0.233,0.309]	[0.223,0.309]	⑦	0.255	[0.211,0.241]	[0.188,0.256]
⑧	6.000	[4.510,5.784]	[4.389,5.860]	⑧	6.000	[4.371,5.851]	[3.680,6.438]
⑨	0.649	[0.623,0.644]	[0.620,0.644]	⑨	0.643	[0.621,0.644]	[0.606,0.647]
⑩	1.799	[1.218,1.753]	[1.178,1.793]	⑩	1.301	[1.097,1.636]	[0.828,1.802]
⑪	8.757	[8.721,10.517]	[8.534,10.661]	⑪	11.710	[10.613,11.743]	[9.960,12.719]

表 6.8 推理解与实际车的值的比较(之二)

形态要素	根据照片测得的实际车的尺寸	推理解		形态要素	根据照片测得的实际车的尺寸	推理解	
		必然解 [下限,上限]	可能解 [下限,上限]			必然解 [下限,上限]	可能解 [下限,上限]
①	0.320	[0.313,0.321]	[0.310,0.322]	①	0.282	[0.282,0.298]	[0.281,0.308]
②	0.275	[0.255,0.290]	[0.259,0.294]	②	0.291	[0.289,0.308]	[0.286,0.319]
③	0.135	[0.110,0.135]	[0.106,0.136]	③	0.071	[0.057,0.091]	[0.058,0.116]
④	28.615	[30.041,33.974]	[29.611,34.016]	④	24.052	[24.151,26.875]	[24.154,28.878]
⑤	33.477	[24.900,29.040]	[24.694,29.605]	⑤	18.147	[16.074,19.788]	[13.033,23.969]
⑥	0.078	[0.080,0.093]	[0.075,0.098]	⑥	0.056	[0.053,0.078]	[0.045,0.092]
⑦	0.255	[0.222,0.310]	[0.213,0.307]	⑦	0.254	[0.208,0.241]	[0.179,0.260]
⑧	5.000	[4.131,5.615]	[3.908,5.722]	⑧	2.000	[1.459,3.499]	[1.280,4.966]
⑨	0.630	[0.576,0.603]	[0.576,0.608]	⑨	0.648	[0.618,0.645]	[0.602,0.645]
⑩	1.180	[1.210,1.868]	[1.157,1.934]	⑩	0.671	[0.669,1.110]	[0.441,1.460]
⑪	10.383	[9.151,10.912]	[9.028,11.120]	⑪	7.943	[8.104,9.589]	[7.198,10.415]

表 6.9 未知设定数据的推理结果

形态要素	推理解	
	必然解 [下限,上限]	可能解 [下限,上限]
① 全高/全长比	[0.301,0.314]	[0.284,0.332]
② 前长/全长比	[0.286,0.306]	[0.287,0.300]
③ 后长/全长比	[0.107,0.135]	[0.078,0.152]
④ A 柱的倾斜度	[28.283,30.633]	[29.281,32.521]
⑤ C 柱的倾斜度	[17.781,26.560]	[17.757,38.411]
⑥ 保险杠突出度	[0.037,0.073]	[0.027,0.082]
⑦ 保险杠厚度	[0.207,0.267]	[0.141,0.251]
⑧ 流线	[3.874,7.486]	[3.525,7.317]
⑨ 腰线高/全高	[0.614,0.642]	[0.614,0.642]
⑩ 腰线倾斜度	[1.161,2.189]	[1.185,2.302]
⑪ 前箱的倾斜度	[10.553,12.306]	[9.588,10.866]

印象用语的数值数据:

- | | |
|---------------|-----------------|
| ① 先进性感觉:0.255 | ⑤ 知能性:0.368 |
| ② 鲜明感:0.474 | ⑥ 正统魅力:0.778 |
| ③ 漂亮感:0.625 | ⑦ 成人的稳重魅力:0.677 |
| ④ 简炼感:0.238 | |

这里 A 的各形状标识数据是把表 6.9 的 (1)~(11) 的形态要素用图 6.16 所示的小,中,大三个隶属函数评价的各个等级,是 11×3 33 维向量。 B 的印象用语为表 6.9 下部的七维向量。

关系 R 与联想存储中式 (5.22) 的联想矩阵 M 的运算基本相同,通过向使用实际数据 A, B 对的模糊关系扩展,得到 33×7 的矩阵。而且,为了简化由印象用语 B 得到形状标识数据 A 的逆推理,使用神经网络实施对其推理结果的学习。因此,双向推理就成为可能。

为了验证获得的模型的有效性,使印象用语的“鲜明度”变化,求得的对车形状数据的结果如图 6.17 所示。图中的双线表示推理必然解的上限和下限。

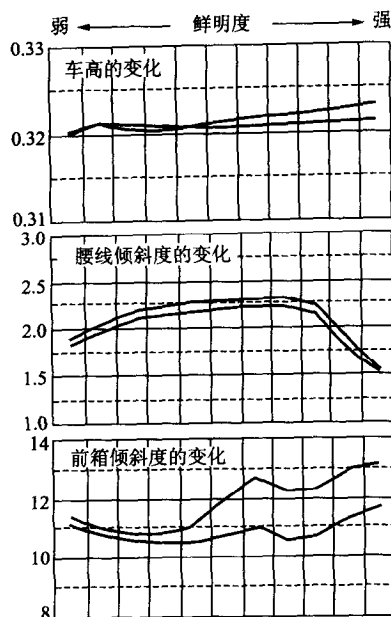


图 6.17 设定数据的验证图

这里得到的汽车设计的评价模型中，可以看出，“鲜明度”的一个印象，不仅对单独一台车的形态要素有影响，而且表现出形态要素之间的相互关系。这些结果，是专家确信的结果，并且也是能够验证人的评价模型的复杂程度的结果。

第 5 章中说明的模糊联想推理中，尽管简单，但也能够构筑评价模型。联想推理中，式 (6.2) 的 A, B 分别被式 (5.19) 和式 (5.20) 的 A, B 替换，式 (6.2) 的关系 R 被式 (5.22) 的 M 替换。

还有联想推理中，可以从 A 侧向 B 侧和从 B 侧向 A 侧两个方向推理，但正确表示本节的模糊关系的上限值，下限值却很难。像表 5.1 中说明的那样，把伴随推理时反响动作的变化的推理法进行简化，通过得到与 AND 条件和 OR 条件相近的下限值，上限值，可以求得简易的结果。

6.2.4 汽车领域中的应用

下面是模糊理论在汽车操纵中驾驶者的感官评价模型的应用举例。图 6.18 所示的是在驾驶者-车辆的闭环控制系统中，驾驶者驾驶车辆时的感官评价模型，即与驾驶者评价驾驶车辆难易程度的感性相关联的模型的实现，为了简化模型，着眼点放在图 6.19 所示的跟随车辆 A 的车辆 B 的驾驶者的加减速操作上。

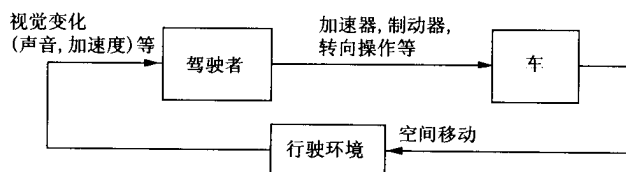


图 6.18 驾驶者-车闭环控制系统

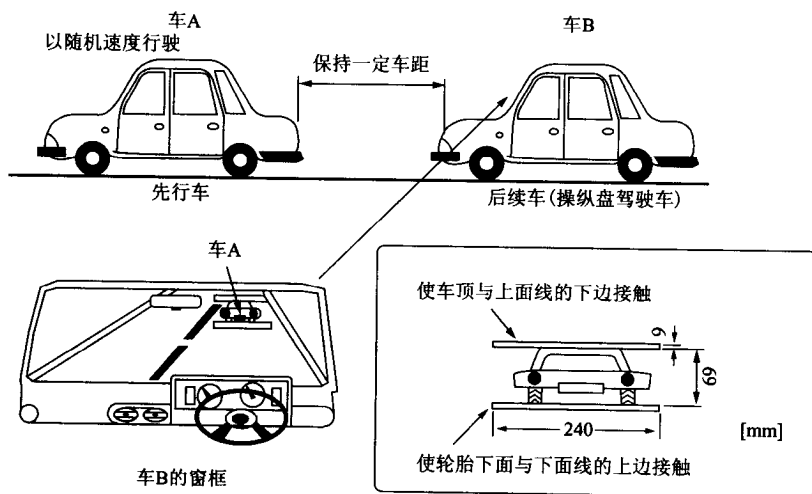


图 6.19 跟随先行车实验概要

图中 建立在车辆 B 中能看清楚车辆 A 与车辆之间的距离关系。
图 6.20 所示的区间中，求驾驶者的加减速操作模型。把区间按其状况进行 k 分割，在每个区间根据自己回归移动平均 (ARMA) 模型¹⁾ 进行鉴定。这里 θ 为 ARMA¹⁾ 模型中的系数向量。

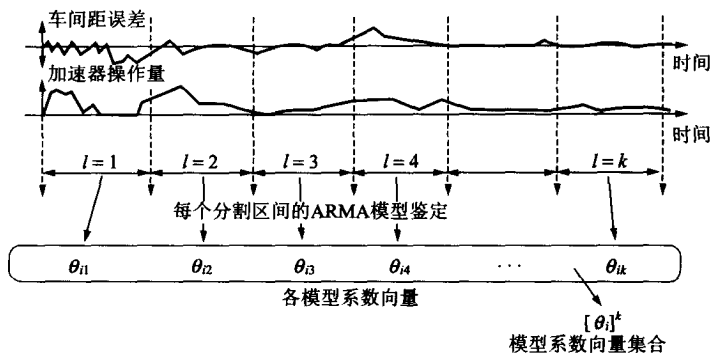


图 6.20 每个数据分割区间的模型鉴定

由于车辆惯性，驾驶者根据经验知道，车速在一定时间常数下是变化的，因此考虑以前加减速操作状况的同时，要估计车间距的误差变化。

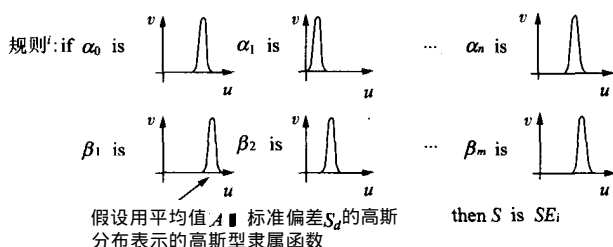
ARMA 模型是对驾驶者的车间距离偏差和加减速操作量联系起来的适当模型。这个模型的系数向量 θ 根据移动平均项的系数 $\alpha_i (i=0, 1, \dots, 10)$ 和自己回归项的系数 $\beta_j (j=1, 2, \dots, 10)$ 定义为

$$\theta = [\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{10}, \beta_1, \beta_2, \dots, \beta_{10}]^T \quad (6.3)$$

使用 ARMA 模型 在 4 位驾驶者中对被认为是通用的操作模型进行了鉴定。同时，通过问卷调查求得 5 段评价的感官评价价值，其结果表示在表 6.10。根据这个鉴定了的操作模型的系数和感官评价价值，由图 6.21 所示的模糊规则得到了感官评价模型。

1) 参看本系列书中的《ARMA 系统与数字信号处理》一书。

图 6.21(a) 中 μ_i, β_j 为操作模型的系数值的输入, $G\{\cdot\}$ 为图 6.21 (b) 中所示的用平均值 A_v 标准偏差 S_d 的高斯分布型双弯曲函数对输入值 α_i, β_j 进行评价。S 表示感官评价的输出。本图是 4 速固定行驶时的感官评价模型。根据实验, 由 4 人的操纵盘 A, B, C, D 来建立关于 4 速固定和 2 速固定的感官评价模型。



(a) 模糊评价知识与隶属函数

规则¹: if α_0 is $G\{A_v=7.23, S_d=1.72\}$ and α_1 is $G\{A_v=10.64, S_d=3.17\}$ and α_2 is $G\{A_v=0.34, S_d=1.94\}$ and α_3 is $G\{A_v=2.77, S_d=2.98\}$ and α_4 is $G\{A_v=0.14, S_d=0.67\}$ and α_5 is $G\{A_v=2.30, S_d=2.14\}$ and α_6 is $G\{A_v=0.64, S_d=1.42\}$ and α_7 is $G\{A_v=1.31, S_d=0.41\}$ and α_8 is $G\{A_v=2.53, S_d=0.26\}$ and α_9 is $G\{A_v=3.18, S_d=2.13\}$ and α_{10} is $G\{A_v=1.11, S_d=0.62\}$

and

β_1 is $G\{A_v=0.53, S_d=0.57\}$ and β_2 is $G\{A_v=0.03, S_d=0.17\}$ and β_3 is $G\{A_v=0.18, S_d=0.06\}$ and β_4 is $G\{A_v=0.02, S_d=0.02\}$ and β_5 is $G\{A_v=0.02, S_d=0.02\}$ and β_6 is $G\{A_v=0.06, S_d=0.03\}$ and β_7 is $G\{A_v=0.02, S_d=0.04\}$ and β_8 is $G\{A_v=0.04, S_d=0.02\}$ and β_9 is $G\{A_v=0.03, S_d=0.03\}$ and β_{10} is $G\{A_v=0.02, S_d=0.01\}$

then S is 3

但是, $G\{A_v, S_d\}$ 表示用平均值 A_v 标准偏差 S_d 的正规分布函数近似的高斯型隶属函数

(b) 实际操纵盘 A 的 4 速固定行驶时的模糊评价知识

图 6.21 官能评价模型的一个例子

在表示感官评价模型的模糊规则的推理中, 需要对第 5 章的推理方法进行改进。因此, 如图 6.22 所示, 要用适合于全局的推理。

一般, 把现有的模糊推理用于评价问题时, 因规则 if 部的一部分的低匹配影响, 得不出全体匹配的全局结论, 结果使模糊性扩大。

表 6.10 本系统的官能评价判断预测

后续车 驾驶者	4 速 固 定 行 驶	3 速固定行驶	
		实际评价	基于模型 的推定
操纵盘 A (25 岁 男性)	3	5	4.8
操纵盘 B (31 岁 男性)	3	4	4.2
操纵盘 C (30 岁 男性)	3	3	3.1
操纵盘 D (31 岁 男性)	3	5	4.6

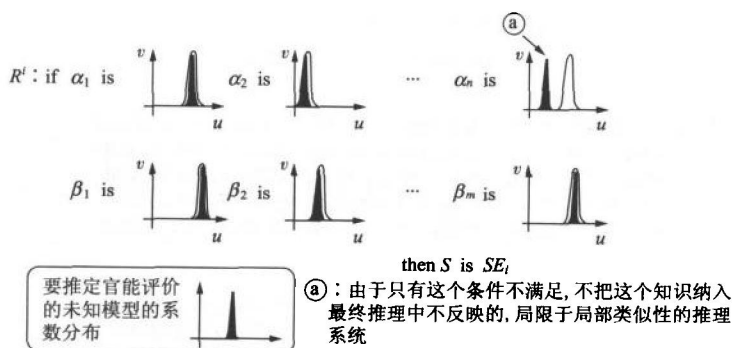


图 6.22 关于全局适应性的推理

考虑到这点, 在这里使用了图示的可得到全局结果的推理。模糊联想推理中, 同样也能够全局推理。像表 5.1 说明的那样, 通过增加联想的反响动作, 可以回避部分不一致, 从而得到全局一致的结果。

第7章

人类的并行处理模型 与智能信号处理

CECECECECECE

- 7.1 联想存储的智能信号处理
- 7.2 联想存储的模糊联想处理
- 7.3 新型智能信号处理

神经网络和模糊理论，是对以人为代表的诸多生物的适应功能，识别功能，学习功能和判断功能等进行深入考察之后导出的。因此，模拟生物的高度信息处理功能，具有实现智能信号处理的极大潜在能力。作为其应用示例，本章将对用联想存储的智能信号处理和模糊联想处理予以说明，并对今后的智能信号处理予以展望。

7.1 联想存储的智能信号处理

人的智能功能有记忆、识别、思考、想象、推理、判断等，但最基本的可以说是记忆功能。对于高度的智能活动，如果没有以前学习的知识和规则是不可能实现的。这里将叙述使用神经网络的联想存储的智能信号处理。

首先说明着眼于人的记忆的情节联想存储；然后介绍以更加柔性的回想为目标考虑混沌（chaos）的导入。这样就产生了混沌神经网络的联想存储，我们将对应用混沌神经网络联想存储的智能信号处理进行阐述。

7.1.1 情节记忆

人的记忆可以按图 7.1 进行分类。首先分为长期记忆和短期记忆两大类。长期记忆是指以文字的形式长期被记忆，而短期记忆是指感觉和知觉，例如知识的暂时记忆等只在比较短的时间内被保持的记忆。长期记忆又可以分为对有关事实进行记忆的声明记忆（declarative memory）和有关声明记忆的应用方法的知识及有关各种顺序等的知识进行记忆的程序性记忆（procedural memory）。

声明记忆又被分为情节记忆（episodic memory）和意思记忆（semantic memory）。情节记忆是以表示时间、地点、事件为中心的记

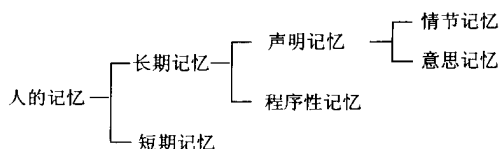


图 7.1 人类记忆的分类

是基于时空间的前后连贯的场景和所发生的事情的记忆。意思记忆是作为普遍事实的知识的记忆，不需要时间、地点等有关时空的信息。

因此从回想的观点看，情节记忆时为“想起……”的形式，意思记忆为“知道了……”的形式^[1]。

互联型神经网络的特征性功能是联想存储。但是，这只是存储单一模式，即存储片段性知识，可以认为是进行简单意思存储的模型。

这里要介绍通过扩展 BAM 而能够处理时间概念的情节联想存储模型。

7.1.2 神经网络的情节联想存储

应用神经网络构成情节联想存储时，如何表示时间概念成为一个重要问题。在人的身体中，发生节奏的部分有好几个地方^[2]，于是，我们可以考虑用工学的方法产生某种节奏，然后用其来表示时间概念。

我们用由二进制信号构成的模式（符号）作为节奏。这是因为考虑到目前全部神经网络的联想存储处理都是二进制模式。还有，这里用 PN(Pseudo-random Number) 符号作为符号。PN 符号由于有下述特征^[3]，正如后面说明的那样，通过将其巧妙应用，就可以使时间概念的表达变得简单易行。

(1) PN 符号的特征让系列进行巡回，用所有的状态与每个项比较，一致项的数目和不一致项的数目最多只相差 1 个。

下面所示的符号是周期为 15 的 PN 符号的一个例子。

010011010111100

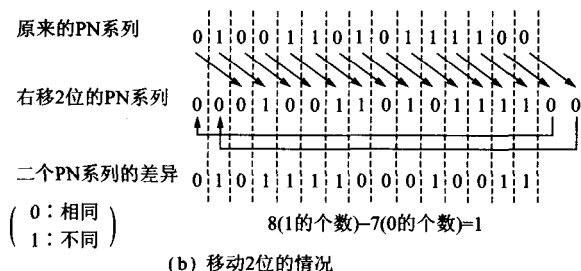
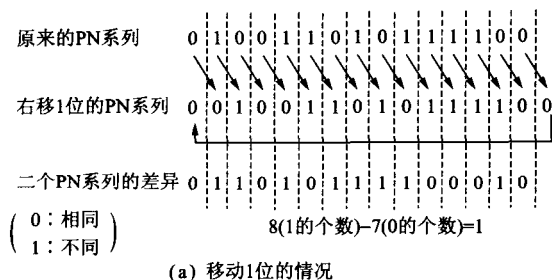


图 7.2 PN 符号的性质

图 7.2(a) 为在任何位置移动 1 位与原来的符号进行比较，各个项一致时记为 0 不一致时记为 1 可见 不一致的数只多 1 位。

同样 在进行 2 位移动的符号比较时，如图 7.2(b)所示得到了相同的结果。也就是说，PN 符号中，让原来的符号巡回移动，对形成的全部模式的相关是相同的。利用这个性质，一边使 PN 符号巡回，一边进行使用，就可以表示时间的概念即顺序。

图 7.3 就是考虑了以上 PN 符号的特征的神经网络的情节联想存储^[4]。情节记忆的最大特征是有时间概念，一侧的层处理时间概念 另一侧的层处理“在哪儿”和“什么”。

如图 7.4 所示 使用 MAM 作成 3 层，用各层也能够处理“何时”，“何地”，“何事”，但这里为了简单起见 采用 2 层模型。

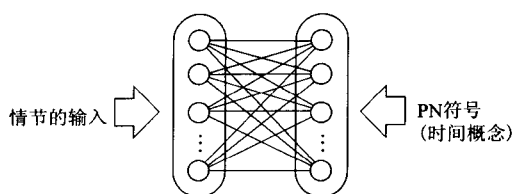


图 7.3 神经网络情节的联想存储

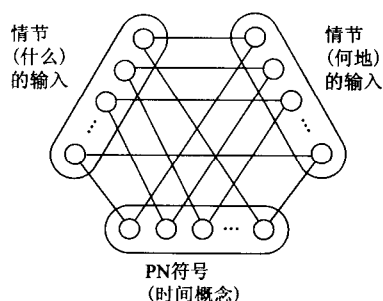


图 7.4 3 层结构的神经网络的情节联想存储

首先说明存储过程。

图 7.3 中，在左侧层输入相当于“何地”和“何事”的模式。在右侧层输入基于 PN 符号的移动的序关系。现在，做两个系列的情节，如图 7.5，分别用希腊字母的大写和小写系列表示。对于各系列（情节），用不同的 PN 符号使其巡回的同时进行存储。

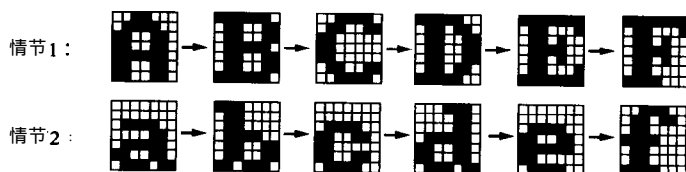


图 7.5 被情节联想存储记忆的两个系列（情节）

BAM 中，由于进行内容的检索，对于噪声和不完全输入具有鲁棒性。因此，即使在图 7.6 所示的被噪声污染的模式输入的情况下也能够据此回想最初的正确 PN 符号。所以，使其符号一个一个巡回，就可以将存储的情节按顺序回想。

如在前面说过的那样，随着将其进行多层化，可以更细地表示存储项目。通过想办法对回想机构进行控制，也可以处理抗噪声的模糊性信息，因此极有可能成为能进行柔软知识处理的新的范例。

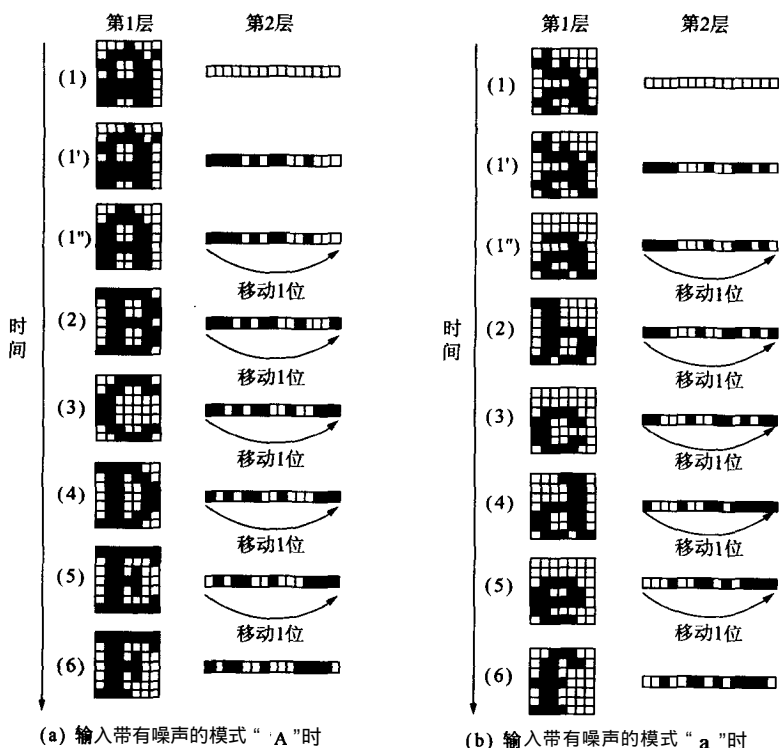


图 7.6 情节联想存储的回想示例

7.1.3 混沌神经元

上一节的情节存储是按顺序回想存储模式的。对于人来说，产生每个想法时的回想顺序是不定的。经常是由一个刺激开始，一个接一个回想各种事情。考虑到更接近于人的信息处理方式，这样考虑回想顺序不确定的神经网络就十分有价值。

为了产生这样的现象必须进行平衡点间的变化。作为能够实现的方法，混沌概念受到了关注。混沌是非线性系统中产生的现象，具有确定性但却是不可预测的这样的奇妙特征。混沌被发现于生物诸多的脑神经系统中，而且，暗示着混沌在生物体中起某些本质作用的可能性^[5,6]。

这里，介绍使用混沌神经网络的联想存储^[5~8]。作为在神经网络中产生混沌的方法，有以下几种：

对与一个神经元结合的权值数增加限制，来强制性使网络不稳定的方法^[6]；

把神经元的动作方程式做成二次系统，而且将非线性导入的方法^[7]；

应用混沌神经元的方法^[6]。

文献[6]的混沌神经元是考虑来自神经网络的外部输入和内部的神经元之间的信号传输而被模型化的。尽管是引入生理学知识的自然模型化，但是仍能够发生混沌，而且包括全部一般人工神经元模型。

动作方程式用下式表示：

$$x_i(t+1) = f \left[\sum_{j=1}^M v_{ij} \sum_{d=0}^D k_{\text{ext}}^d A_j(t-d) + \sum_{j=1}^N w_{ij} \sum_{d=0}^N k_{\text{feed}}^d x_j(t-d) - \alpha \sum_{d=0}^D k_{\text{rest}}^d x_i(t-d) - \theta_i \right] \quad (7.1)$$

式中 $x_i(t+1)$ ——神经元 i 在时刻 $(t+1)$ 的输出；

$f[\cdot]$ ——神经元的输出函数（连续函数）；

v_{ij} ——来自外部输入 $A_j(t)$ 的对神经元 i 的权值；

k_{ext} ——对外部输入的时间衰减常数 ($0 \leq k_{\text{ext}} < 1$);

$A_j(t)$ ——第 j 个时刻 t 的外部输入;

w_{ji} ——神经元 j 对神经元 i 的权值;

k_{feed} ——来自其他神经元的对反馈输入的时间衰减常数
($0 \leq k_{\text{feed}} < 1$);

α ——常数 ($\alpha \geq 0$);

k_{rest} ——神经元的不对应性的时间衰减常数 ($0 \leq k_{\text{rest}} < 1$);

θ_i ——神经元 i 的阈值。

上式右边的函数 $f[\cdot]$ 中的第 1 项为来自外部的输入, 第 2 项为来自其他神经元的输入, 第 3 项为考虑其神经元自身不对应性的项, 第 4 项为其神经元的阈值。

与普通神经元模型的很大不同点是神经元的输出函数 $f[\cdot]$ 是连续函数。另外, 式 (7.1) 若用递推公式, 可以大幅度减少计算量。

7.1.4 混沌神经网络的联想存储

若考虑人的智能信息处理方式, 关于记忆, 还有一点必须考虑。对于人来说, 比如一听到“红的”就联想起“苹果”、“太阳”和“消防车”等各种事情。也就是说, 记忆是“1 对多”的结构。

然而, 对于神经网络的联想存储进行“1 对多”的存储非常困难。这是因为, 应该回想的数个模式间产生干涉, 生成复杂的变形模式。

这里通过增加附加信息把学习模式近似地进行“1 对 1”的方法, 来解决“1 对多”的问题, 具体方法在后面叙述, 还考虑到前项叙述的回想顺位的不确定性问题, 所以应用混沌神经元。

根据以上考虑就构成了混沌双向联想存储 (CBAM: Chaotic Bidirectional Associative Memory)^[9]。其结构见图 7.7。

CBAM 中, 如图 7.7 所示, 一侧的层 (x 层) 分为关键输入和连贯两部分。这里只有连贯部分的神经元是混沌神经元, 其他用 BAM (参见 3.3 节) 的普通神经元模型。

图 7.8 显示了 5 组学习模式。对于关键输入 B 乌鸦和企鹅与之对应 对于关键输入 M 老鼠和猴子还有猫与之对应。可以看出 分别是“1 对多”的学习模式，但由于连贯部分不同，近似地成为“1 对 1”的组合。

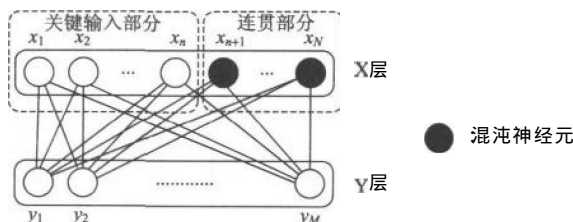


图 7.7 混沌双向联想存储的结构

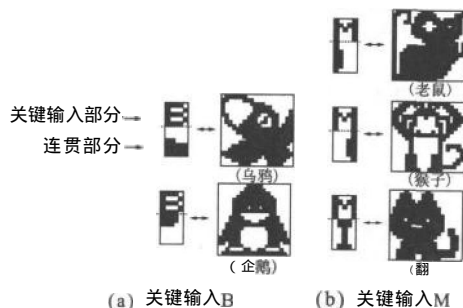


图 7.8 5 组的学习模式

用 BAM 的普通学习方法（相关学习）学习这 5 组时，回想结果的例子表示在图 7.9 中。

这里，与老鼠、猴子和猫对应的 M 在时刻 $t=0$ 时对 X 层输入。在 $t=1$ 时，这些重叠模式虽然在 Y 层被表示，但是连贯部分的神经元状态随混沌而转移，在 $t=3$ 时老鼠被回想，这一直持续到 $t=6$ 。然后可以看出，在 $t=8\sim11$ 时是猫，在 $t=15\sim20$ 时是猴子，分别被回想。再往后重复进行状态转移的同时老鼠，猴子，猫的模式被回想。

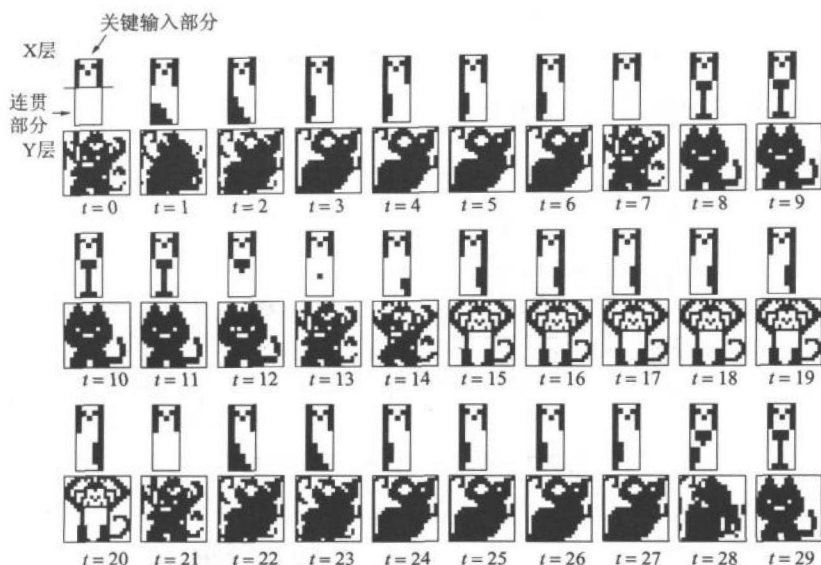


图 7.9 混沌双向联想存储的回想示例

这样，回想模式一个接一个变化，是因为网络状态不是停留在存储的稳定平衡点，而是在包含存储模式和其重叠模式，以及反转模式的各种模式间进行非周期性回想。

另外，即使用混沌神经元，参数没有得到很好调节的话，也不会出现这种现象。必须要注意，混沌现象的发现与参数的依存性很高。

我们在必须解决某个问题时，是在全面考虑各种事情想出解决方法之后，进行综合讨论最后才得以实行。要想出解决方法，“思路”特别重要。在思路中，有这样的情形，即有时乍一看认为与问题没有一点儿关系，但随着类推和直觉会发现划时代的解决方法。“思路”的实现，用基于逻辑的以前的人工智能的研究方法是极其困难的。

这样的“思路”、“偶然想法”、“类推”、“直觉”等在神经网络的联想存储模型中，单靠增加噪声是不可能实现的。这是因为，噪声太小不能产生构思的飞跃，而把噪声放大，只不过是随机回想各种事情。

如果用混沌神经网络，模式的周期回想，或者各种事情的动态回想，通过参数的调整将会成为可能。前者相当于思考的环绕，后者相当于思考的流程。从以上看可以预计，在工学上实现人的智能信息处理功能，混沌将起重要的作用。

7.2 联想存储的模糊联想处理

近几年，美国提出了信息高速公路的构想，信息基础设施在飞速建设中。与此同时，因特网、可视化等办公和家庭中高速双向多媒体正在逐步实现¹⁾。

在日本，IMS (Intelligent Manufacturing System, 智能制造系统) 项目以生产系统的新型自动化和智能化为目标开始实施。今后，还会开展包含有这两者结构的智能工业系统(IIS)的实现研究。

IIS 是这样的系统，如图 1.10 中说明的那样，信息基础设施作为基础，由少数人和多数智能机械构成，为了达到目标，由人来进行粗略判断，然后给机械下达指示并获取信息。

也就是说，是由人来分担大型的智能动作，信息基础设施上的智能代理机是辅助人的智能动作的系统。

这个智能代理机，由基于层次性知识的软计算，通过与人的交互中接受指示命令和理解其性质上的微妙差别，把来自外界的数值信号向与人亲和性高的符号进行变换等，来实现与人及外界的对话。

如图 7.10 所示，作为智能代理机的必要功能，是以软计算为基础的 3 个功能，①多形态接口功能，②宏观使用定性指示的知识精练功能，③谋求自律性构思的功能。

通过使用这样的智能代理机，就能够实现自然用户接口 (NUI)。本节，将解释作为其第一步的应用智能代理机与人进行宏观对话，给

1) 参看本系列书中的《多媒体与数字信号处理》一书。

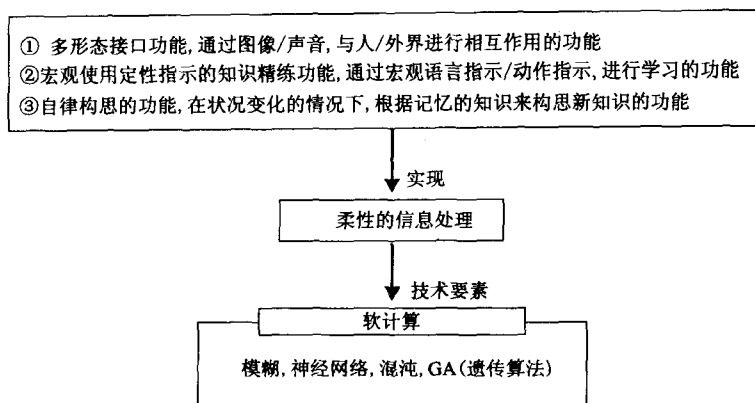


图 7.10 智能代理机中必要的功能

人以支援的智能信号处理系统的具体例子。

对于 的功能, 是以把现实世界的数值信号向宏观标识和其性质的细微差别进行变换的模糊信号处理为基础, 并加以应用, 这使得诸多信号之间的相互变换和信号统和就成为可能。

另外, 作为实现本节中 和 的功能的一个例子, 采用图 7.11 所示的基于直觉的代理机模型。

这个基于直觉的代理机模型是, 为了实现智能系统, 对作为其层次性知识模型而为人所知的 J. Rasmussen 的智能层次模型进行扩展, 以联想存储为基础的直觉处理, 附加了自下而上处理和自上而下处理的模型。

这个层次模型 由 (a)局部循环, (b)微观知识, (c)宏观知识等 3 层构成。(a)的局部循环, 输入来自外界的数值信号, 形成对外界反应的数值信号进行输出的循环。在这一系统中构成反馈的情况下, 可以应用模糊控制。(b)的微观知识, 产生一个动作中的一系列顺序和目标的模式。(c)的宏观知识, 把握人的意图, 进行动作顺序的计划和对其进行评价。

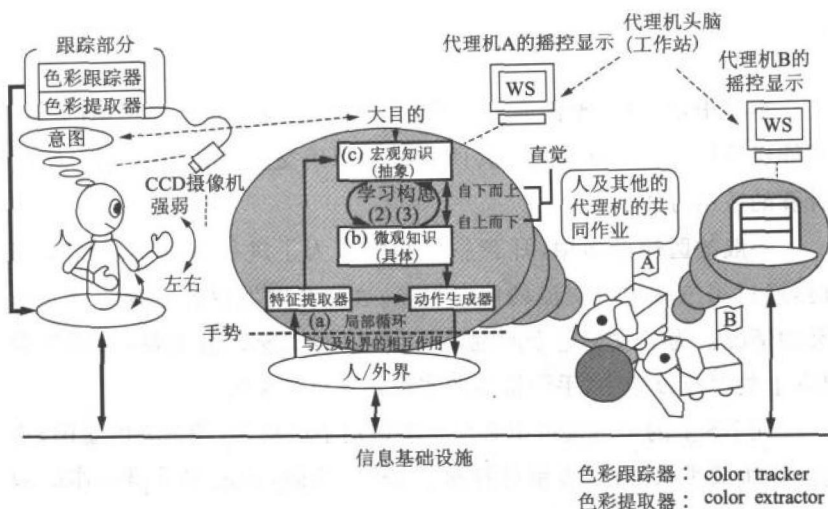


图 7.11 基于直觉的代理机模型

这里，以图 7.10 中的知识简化功能的实现为例，对 7.2.1 项中用手语和人对话的代理机，根据 7.2.2 项中根据人的动作指示来理解人的意图的动作进行简化，理解人的动作指示的智能机器人等予以介绍。还有，在这些实现例子中，对来自人的动作指示的数值信号由联想存储的直觉处理进行评价，理解人的意图的方法等同时予以介绍。

这些都是“基于直觉的智能代理机和人的相互作用”的一个模型，我们将在 7.2.1 节中解说“人-智能代理机-人”系统，在 7.2.2 节中解说“人-智能代理机机器人”系统模型。

在 7.3 节中作为图 7.10 中的构思功能的具体例子，将展示对 7.1 节介绍的联想存储的混沌回想进行扩展的知识构思法^[14]和智能代理机机器人的构思功能的路径计划。这个例子显示的是“人-智能代理机机器人”的模型。而且，在这个模型中也对并行处理和进化模型的应用的可能性予以说明。

7.2.1 手语交互式接口

本项中，采用基于直觉来实现智能代理机与人的相互作用的“人-智能代理机-人”系统模型。具体地是用手语交互式接口^[15]来说明软件计算的适用方法。

一般的医疗设施中，耳聋患者和医生为了进行病症传递等交流时，通过第三者进行交流的情况很多，这样会发生侵害患者的个人隐私的情况。为了改善这个问题，手语接口的实现就很重要，特别是在IIS上的医疗设施中，手语接口的实现是不可缺少的。

在IIS上的手语接口中必须有 识别手语动作（理解人的意图）功能 ②生成手语动作（表示计算机的思想）功能，追加手语动作的知识（学习新的手语动作）功能。

在 的功能实现中，手语接口只用状况（前后连贯）依存和手的动作的识别来进行处理。而且，为了识别非特定者的手语动作，①的功能用模糊标识匹配的方法来实现。

这个动作识别方法中，（a）把作为来自跟踪装置的时空间模式来表示的人的动作数据，替换成定性状态的模糊标识，即标识和给其标识赋予特征的隶属函数的转移，构成了评价动作的模糊规则。

然后，（b）把依存于状况的知识，用与概念模糊集合相同的方法，进行阶层性规则化构成。

根据联想推理把以上两种知识进行统和，来进行动作识别。

联想推理根据联想存储来实现 具有“模糊性的控制”、“推理的双向性”、“多段推理的实现”等三个特征。其中 根据“推理的双向性”这一特征，使将手语动作中部分信息和由一系列动作的全体信息进行融合的推理成为可能，并且能够识别非特定者的手语动作。

还有 根据“模糊性的控制”、“多段推理的实现”这样的特征，可以实现规则的多层化。而且，通过把依存于状况的知识（手语动作的复杂性和语顺）进行规则化构成，可以识别很多手语动作，即使是相似的

动作，也能够对意思不同的手语动作予以区别。

的功能主要是应用联想推理的“推理的双向性”特征，由联想存储进行存储，把为了某个手语动作评价的规则进行反方向应用来实现。用计算机，把手语动作手的轨迹点（识别的情况下）进行反方向推理，将其轨迹生成原来手语动作的动画。

③利用联想存储的学习功能。本节中用 ABL (Activation Bidirectional Propagation Learning) 法。这个 ABL 法可以生成手语动作评价规则，由联想存储进行存储，然后追加到某个已有的手语动作评价规则中。所以，经过数次学习，对其规则进行修正，就可作为真规则（和已有规则有同等信赖度的规则）。据此，可以学习用于新的手语动作评价的知识。

1. 手语接口中的智能层次模型

手语接口的实现中，如图 7.12 所示 是对 Rasmussen 的层次模型进行了扩展。这个层次模型，是为了进行智能控制，把人的行动分为三个层次构造的分类模型。

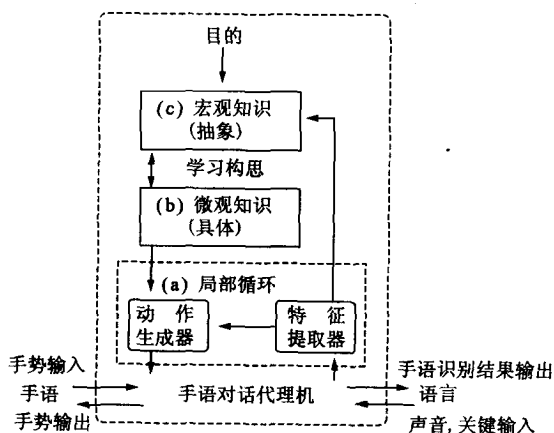


图 7.12 手语对话代理机

这个模型有宏观知识，微观知识，局部循环三个等级构成。宏观知识是最上位，局部循环是最下位，上位处理的是抽象信息。因此，根据自上而下处理和自下而上处理相融合的联想的直觉处理，来进行识别和生成等的推理。

根据这个智能层次模型来构筑手语接口，实现手语动作识别功能，手语动作生成功能和手语知识学习功能。用这个手语接口，进行自上而下处理的联想推理，生成手语动画。并且进行自下而上处理的联想推理，来进行手语的识别和知识的学习。

2. 系统构成

如图 7.13 所示，手语接口系统由手语动作识别、手语动作生成和手语知识学习等三部分构成。

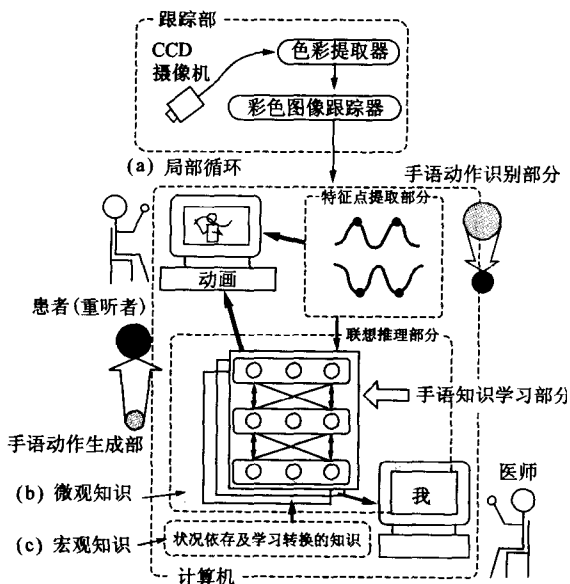


图 7.13 系统构成

在手语动作识别的跟踪部分，把手的位置用二维坐标系中的时系列信号算出。在特征点提取部分，通过时系列波形检出山或谷作为动作的特征点，把模糊标识分配给其特征点。如果手语动作结束，就根据检出的特征点和手语动作的复杂性和语顺实行联想推理，进行动作的识别。

在手语动作生成部分，利用联想推理部分的手语动作评价知识。用计算机进行联想推理，得到手语动作的特征点。应用计算机的图形功能，生成把其特征点平滑连接的动画。

在手语知识学习部分，当进行现有的知识中没有的手语时，为了学习手语知识 生成新的手语动作评价用规则 由联想推理部分存储 并追加到某个现有的手语动作评价用规则中。所以，通过数次学习，对其规则进行修正，就可以作为真规则（和已有规则有同等信赖度的规则）

下面，对各部分进行详细叙述。

（1）手语动作识别部分

跟踪部分：如图 7.13 所示，由 CCD 摄像机，色彩提取器（color extractor）和彩色图像跟踪器（color image tracker）组成。这里，手的位置用在摄像机视角的二维坐标系中的坐标值算出。然后，把算出的坐标值以 1/60 秒的速度向计算机传送。坐标值的纵方向和横方向都设定为 $\pm 0.6\text{m} \cdot \pm 1000$ 。根据目前的跟踪技术的发展，色彩的识别和坐标值计算可以用硬件实现。因此，把这部分作为前提条件处理。

特征点提取部分：在跟踪部分算出的坐标值作为时系列信号接收。因此，由图 7.14 所示的时系列波形检出作为特征的峰或谷，并配以模糊标识。

人不是以时空间模式来捕捉对象的动作，而是以一连串的动作的视角捕捉。因此，根据时系列波形中的峰、谷为特征，要把这个捕捉方式片段性地进行模拟。而且，由于人存在体格差异，用放在头和膝的位置的手的距离对特征点值的进行正规化，以此作为非特定者的手语动作识别的一个对策。

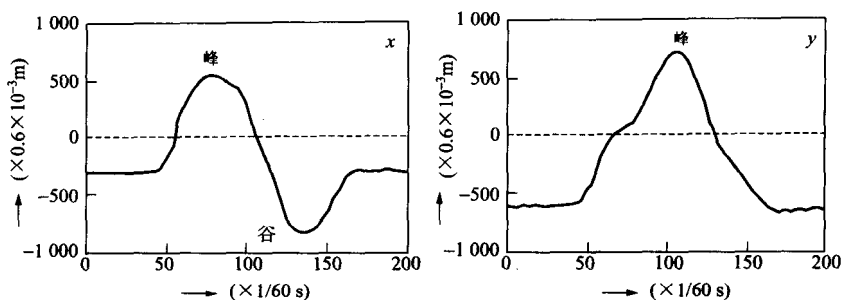


图 7.14 手语“山”的时系列信号示例

联想推理部分：根据所进行的手语动作的词类和手语动作的复杂度，比如一只手的手语和两只手的手语的区分和特征点数等，进行如图 7.15 所示的手语动作分组，对其状况依存的知识进行规则化，并在用于手语动作评价的规则的上位进行构成。

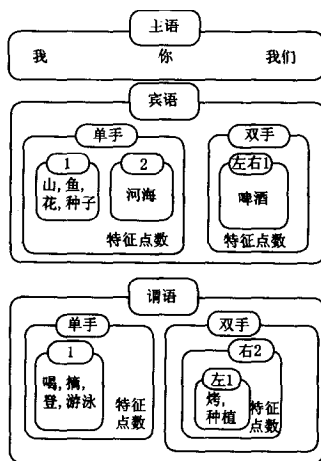


图 7.15 手语动作的分组

因此，联想存储网络，只把手语动作的组数作为下位知识。通过这样的层次性构成，可以识别大多数的手语动作，而且能够区别动作相似而意义不同的手语动作。

首先,根据状况依存规则进行用于手语动作评价的联想存储网络的选择。然后,在特征点提取部分从分配有模糊标识的一系列特征点开始实行联想推理,来识别手语动作。

(2) 手语动作生成部分应用联想推理的“推理的双向性”特征,与识别时的方向相反,使用用于手语动作评价的知识。通过计算机,把手语动作的知识作为向所属的联想存储网络的 then 层的输入值,进行联想推理。因此,在 if 层得到手语动作的特征点。

然后,把这个特征点用二次曲线平滑连接算出手的轨迹。根据其轨迹以 1/60 秒描绘手的位置移动的一个画面。由以上顺序,生成手语动作的动画。

(3) 手语知识学习部分人根据手的动作指示手语学习的时候,由层次型模型的上位规则来进行学习及识别的切换。进行手语学习时,用手语动作学习部分的上位宏观知识,根据在手语动作识别部分提取的数值数据,按照状况依存的规则进行要学习的单词的分组,生成并追加作为微观知识的新的手语知识。然后,通过对其手语知识进行修正,学习用于手语动作评价的知识。

这里通过根据 ABL 法进行知识的生成和修正处理,对作为微观知识的用于新的手语动作评价的知识进行学习。

下面,说明手语知识的生成和修正。

手语知识的生成。新的手语知识生成的步骤如下

第 1 步生成用于手语动作评价的规则。比如对评价表示“喜欢”的手语动作的规则如式 (7.2) 所示。

if x_1 is LIKE, x_2 is LIKE, y_1 is LIKE, y_2 is LIKE

Then word is LIKE (7.2)

第 2 步定义用于评价特征点的隶属函数。以进行手语动作时的特征点为中心(隶属度 1),采用有前后一定幅度的三角形隶属函数。

第 3 步选择存储手语动作的词类和由手语动作的复杂度生成的规则的网络。然后,对其网络的 if 层, rule 层, then 层追加节点。“喜

欢”的情况下，将其追加到词类为谓语，单手手语，特征点数为 1 的组。

第 4 步把追加的节点间的结合权值设定为已有的知识的权值的 $1/2$ 。这个结合权值相当于知识的信赖度。在学习初期阶段，由于对新的知识的信赖度比较低，所以取已有的知识的权值的 $1/2$ 。

手语知识的修正。生成的手语知识是由一个手语动作数据生成的，其信赖度很低。因此，还要根据众多的手语动作数据，修正隶属函数及网络内的节点间的结合权值。

网络内的节点间的结合权值由式 (7.3)～式 (7.5) 求得进行修正。根据这些公式，在识别手语动作后，增加结合的权值，对已经学习的手语知识的信赖度增高，而在未完成识别手语动作时，减少结合的权值，对已经学习的手语知识的信赖度变低。

$$W(new) = \frac{\alpha M(now) - M(old)}{2\alpha} [1 - \delta W(now)] + W(now) \quad (7.3)$$

$$W(now) = W(old) - \gamma [1 - M(old)] \quad (7.4)$$

$$ID = \beta [1 - W(now)] \quad (7.5)$$

式中， W 为网络的 if 层和 rule 层的结合权值 ($0 \leq W \leq 1$)， M 为 if 层中的激活值 ($0 \leq M \leq 1$)， ID 为隶属函数的修正度。 (old) 为学习前， (now) 为学习中， (new) 为学习后的值。 $\alpha, \beta, \gamma, \delta$ 为根据学习的进展程度确定的常数。

隶属函数的修正是在手语动作不能识别时进行。根据式 (7.4) 和式 (7.5) 求隶属函数的修正度。然后按照其修正度，如图 7.16 所示修正隶属函数。

如上所述，通过重复进行手语知识的修正，新的手语知识就成为和已有的手语知识具有同等信赖度的真规则。

3. 实验及结果

(1) 非特定者的手语动作识别把图 7.15 所示的用 16 个单词能够表示的 18 个语句由 5 位被测试者用手语进行图 7.17 是其识别率。

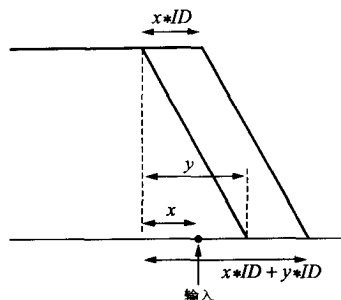


图 7.16 隶属函数的修正

图 7.15 所示的 16 个单词中,含有“我”和“喝”、“鱼”和“游泳”相似的手语动作。在这个识别实验中得到的平均识别率为 82.4%。而且相似的手语动作的“我”和“喝”的识别率分别为 85.3%和 97.6%，“鱼”和“游泳”的识别率分别为 73.9%和 74.7%

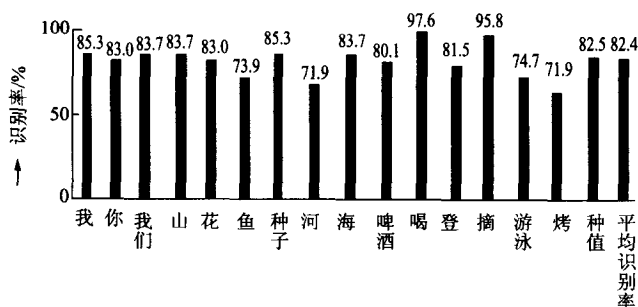


图 7.17 非特定者的手语动作识别率

从以上结果可以得到以下结论,状况依存(手语动作的复杂度和语顺)的知识和用于评价手语动作的知识进行层次性构成,是对防止随着识别语数的增加识别率低而有效的对策,而且能够区别相似手语动作。所以,就验证了这个手语接口有识别手语动作的功能。

(2) 手语动作生成 图 7.18 示出利用计算机生成“我烤鱼”这一手语动作动画时的简单例子。这几个例子是在工作站的显示窗上生成的。从这个结果可以看出，这个接口有生成手语动作的功能。



图 7.18 手语生成的实行示例

(3) 新的手语知识的追加这里，将“喜欢”这个手语动作追加到前述的 16 个单词的手语动作。把“喜欢”追加到词类是谓语，单手手语特征点数是 1 的组。图 7.19 表示的是学习手语动作“喜欢”的时候，随着学习次数变化规则的信赖度的变化情况。

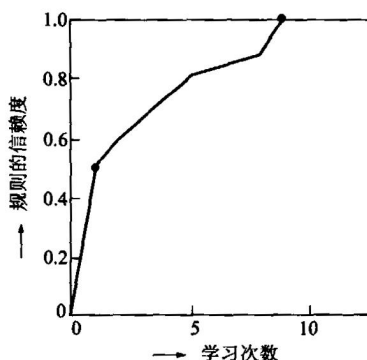


图 7.19 基于学习次数的规则信赖度的变化

另外图 7.20 表示的是根据规则的信赖度的变化产生的属于同一组内的手语动作的识别率的变化情况。这个实验是通过对 5 位被测试者进行测试得到的。

在学习的初期阶段 对用于评价“喜欢”的规则信赖度很低“喜欢”的识别率为 58.2%。随着学习的进行，对评价“喜欢”用的规则的信赖度增高，到成为真规则（学习次数 9 次）时 其识别率为 96.0%。

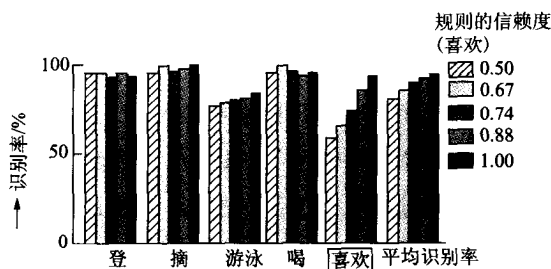


图 7.20 基于学习的识别率变化

随着对评价“喜欢”用的规则的信赖度的增高，其它的手语动作的识别率多少也有点变化，但认为没有多大影响。从以上结果可以看出，评价“喜欢”用的规则在被学习。

所以，验证了这个手语接口有手语知识的追加功能。

7.2.2 智能机器人及其学习

本节所举的基于直觉的智能代理机与人相互作用的具体例子，将展示“人-智能代理机机器人”系统的模型。图 7.11 的 IIS 中，少数人给多数机械下指示是必要的。但是若对多数机械给予过于详细的指示，效率就会降低。

也就是说，人有必要用通常用的宏观信息与机械进行对话。作为能够实现这种情形的 IIS 应用的一个例子，我们对智能代理机机器人

进行介绍。

1. 智能代理机机器人中的智能模型

这个智能代理机机器人也和手语接口一样 根据 Rasmussen 的层次模型构成,由宏观知识,微观知识,局部循环三层组成。

宏观知识具有识别人的动作的知识。微观知识,比如,要实现 8 字形这样的动作,可以由四个圆弧组成的部分的组合形成机器人动作的轨迹,并将其结果送给相当于局部循环的机器人。因此,机器人就进行动作。

智能代理机机器人如图 7.21 所示。以前的机器人,当执行没有识别知识的新指示时,就不能进行识别和执行指示。对于这个新知识的追加,使用也在手语接口中使用的 ABL 法。而且,在人们日常使用的“变大”、“变小”这样的含糊指示下,使动作的轨迹变大 或者在变小的学习法中使用 GIL (Gestural Instruction Learning)法。

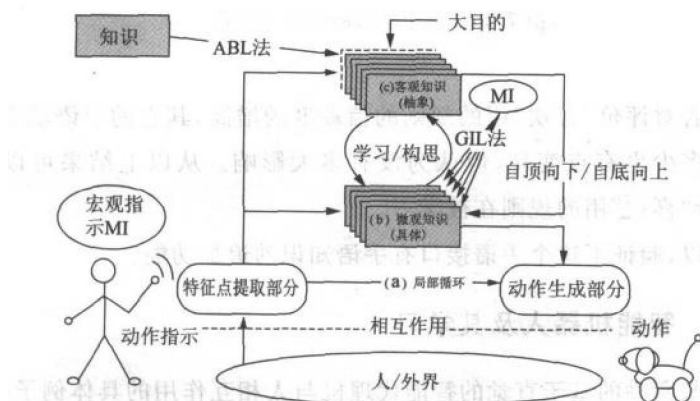


图 7.21 智能代理机机器人

2. 宇都宫犬系统

下面介绍,作为与人协调并执行人的指示的小型移动机器人,宇

都宫大学开发的大老鼠的改进型机器人(以下称为宇都宫犬)。

图 7.22 表示的是识别人的动作指示, 对进行学习的机构予以说明的宇都宫犬系统的结构图。本系统分为图左下部的特征点提取、图中央的联想存储和图右下部的动作生成三个部分。

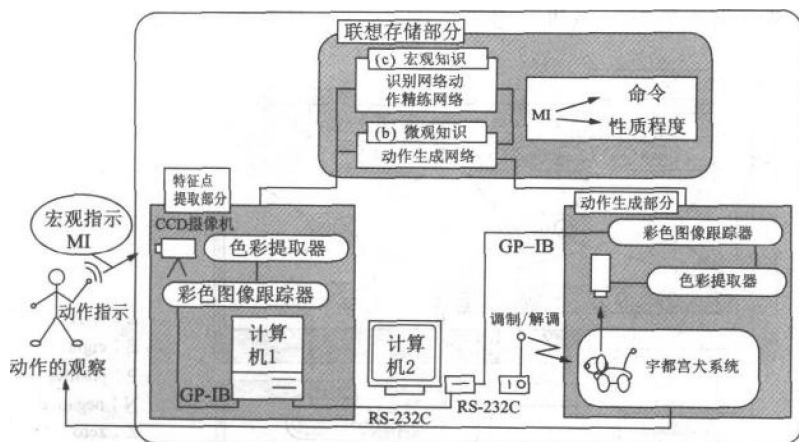


图 7.22 宇都宫犬系统

特征点提取部分：右手戴黄色手套，左手戴红色手套做动作指示。用右手对犬表示 8 字形和心形，左手用来对犬表示 Yes/No。用相当于犬的眼睛的 CCD 摄像机将这个动作取为图像数据。

用色彩提取器从这个图像数据提取出黄色和红色，用色彩图像跟踪器以 1/60 每秒追踪色彩像素的集团移动来提取两手的动作，如图 7.14 所示，用二维坐标系变换成用于进行特征点提取的时系列信号。

这里把 CCD 摄像机，色彩提取器，色彩图像跟踪器汇集，称为采集部。由这里得到的时系列信号用计算机 1 把峰和谷作为特征取出，传送给联想存储部分。

联想存储部分：这部分主要有两类知识。作为宏观知识，从在特征点提取部分得到的右手动作的特征点开始，人做出的指示的意

思 用于识别“变大”、“变小”这样的指示的性质程度的知识。还有微观知识，就是遵循用宏观知识识别的指示和性质程度，将直线和圆弧进行组合来构成动作的轨迹的知识。

除此之外，还有根据左手的动作，识别 Yes/No 进行程序控制的知识。这些部分通过应用模糊联想存储系统来实现，根据由特征点提取部分取出的特征点进行推理。图 7.23 为该联想存储部分的结构。

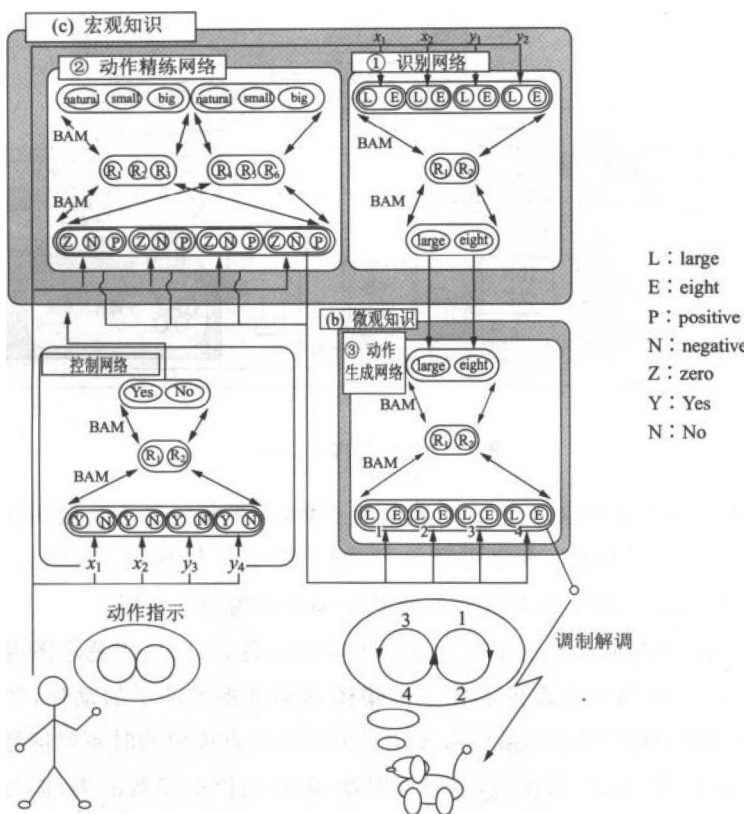


图 7.23 联想存储部分的结构

本系统由对应于宏观知识的识别网络和动作精练网络，对应于微观知识的动作生成网络等三个主要网络组成，此外还有控制程序流程

www.aibbt.com 让未来触手可及

的控制网络。

识别网络把人的指示取为特征点，并理解其指示的意思。动作精练网络把通常的指示和实际所做的指示的差别作为输入，把大小的程度分为 x 轴方向， y 轴方向进行理解。

然后，动作生成网络按照识别网络和动作精练网络的结果，构成犬的轨迹。

动作生成部分：动作生成部分中的跟踪部分基本上与特征点提取部分相同。但是，CCD 摄像机是为了用于把握宇都宫犬自身的位置和方向，用色彩检出器把犬的前部检出为绿色，后部为红色，并算出犬的位置和头部方向。宇都宫犬本身就是使用了在迷路搜索竞赛等中作为的自立移动机器人的大老鼠。

3. 智能代理机机器人的学习

对在宇都宫犬系统中使用的 ABL 法和 GIL 法两个学习方法予以说明。ABL 法和 GIL 法的切换，按照图 7.24 根据左手做的 Yes/No 进行控制。

(1) ABL 法 ABL 法，是制订让犬去做在已有的识别知识中没有的，全新的动作时的情况下，动作识别规则和动作生成规则。而且把已有的某个动作修正成其他的简单指示，为使犬能按照简单的指示行动，也进行动作识别规则的修正。

本学习法按照图 7.24 的学习课程来实行。如图中所示，当不能识别人的指示和显示出对人的指示错误识别时运用 ABL 法。

并且，当规则的信赖值比较低时，为增加信赖值实行 ABL 法。这里的规则信赖值是指学习中的规则的确信度。使用 ABL 法的效果表示在图 7.25。

ABL 法是在新的识别规则的制订和识别规则的修正两方面进行的学习法。该学习法，把新动作的动作识别规则和动作生成规则的节点追加到图 7.23 的识别网络和动作生成网中。而且，在向简单动作指示变更时的动作识别规则的修正中，只对图 7.23 的识别网络实行修正。

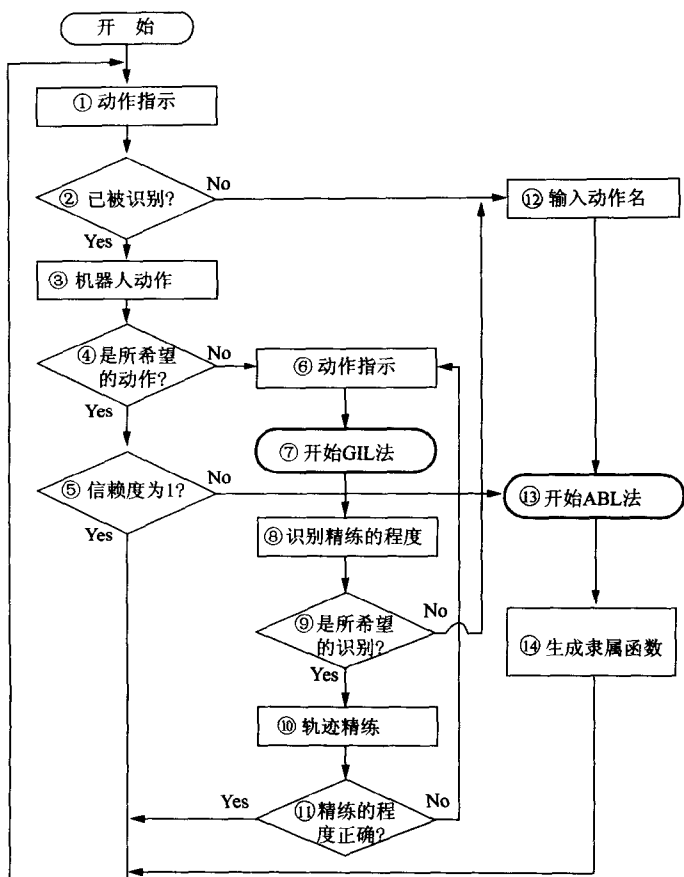


图 7.24 学习过程

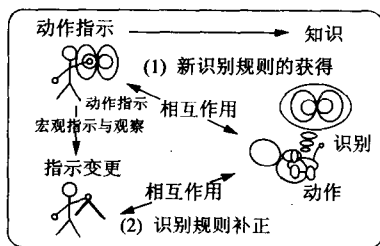


图 7.25 ABL 法

新的识别规则的生成：人在学习新的知识时，一般对刚刚记忆的知识不是充分相信，但是，通过重复进行学习就产生了与已有的知识相同的信赖度。

在新的识别规则生成时，本机器人系统中也考虑了图 7.26 所示的规则信赖值。

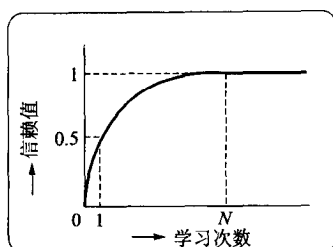


图 7.26 信赖值的变化

信赖值表示为 if 层和 rule 层之间的结合权值。学习中，与手语接口相同 使用式 (7.3)～式 (7.5)。在开始追加规则时，根据输入值做三角形的隶属函数。在这个隶属函数下求隶属度，由式 (7.3) 确定权值。而且，规则的修正用式 (7.4) 求临时权值。

然后 由式 (7.5)修正隶属度最低的隶属函数，根据来自输入的推理求 if 层的隶属度。

最后 由式 (7.3)确定权值。图 7.24 的 ⑭中被赋予了新的识别规则时 如图 7.26 的第 1 次的学习，由于信赖值很低识别规则还不完全。

数次学习后 如图 7.26 的第 N 次学习，新的识别规则就可以得到较高的信赖值。

识别规则的修正：为了让犬做复杂动作，人必须做复杂的指示。因此 如图 7.25 所示，把复杂指示（8 字形等）变换成简单指示（V 字型等）时也用 ABL 法。这时，根据式 (7.5) 的 ID 的值先减少信赖值，然后修正识别规则。

数次修正后 识别规则变化 可以使犬按‘简单指示’做‘复杂动作’。即使在修正后，由于未修改动作生成规则，犬的复杂动作也不变。

(2) GIL 法 在图 7.24 的中动作的轨迹不是所希望的大小时使用 GIL 法。如图 7.27 所示，GIL 法是对人的动作指示的性质程度进行理解，根据其结果进行学习，并由其后的动作反映的学习法。

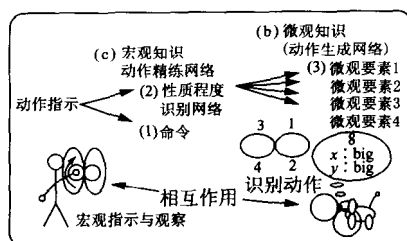


图 7.27 GIL 法

该学习把想在纵或横向变大时的“变大”和变小时的“变小”这些指示作为输入，然后用识别网络识别其指示。在动作精练网络中，把基本的动作指示和“变大”或“变小”的动作指示的差作为输入，得到“大”、“普通”、“小”这样的结果。

把这个结果作为输入，利用模糊联想存储系统的双向性，通过逆推理，得到几倍的纵横值。

然后，在动作生成网络中对与用识别网络识别的结果的动作有关的全部的动作生成规则，用动作精练网络得到的值进行一次修正。

比如，对于纵横向都为大的 8 字形这样的指示，其指示由识别网络识别为指示的意思(命令)是 8 字形 由动作精练网络 作为指示的性质程度 理解为 $x: \text{big}$, $y: \text{big}$ 。根据这个性质程度，为了在纵横向朝大 8 字形变动 对动作生成网络的 8 字形的动作生成规则全部进行简化。

4. 机器人实验

下面是应用上述的 ABL 法和 GIL 法两种学习法的实验结果。

(1) ABL 法的实验动作规则的生成和修正的两个实验。

新的动作指示的生成实验：图 7.28 表示的是作为新的动作指示的大 8 字形的轨迹的学习。

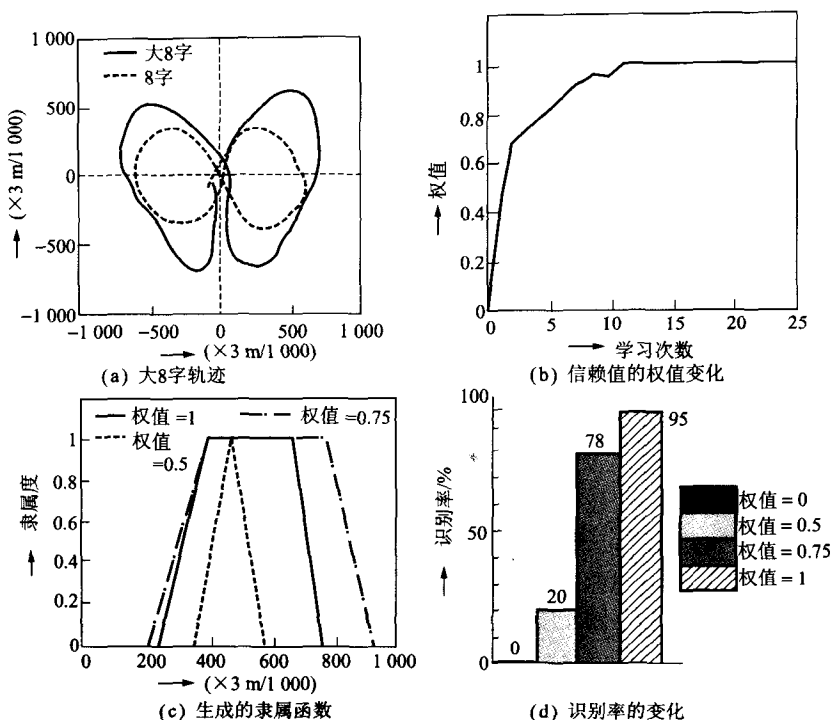


图 7.28 新 8 字的指示的生成

图 7.28(a)是新被学习的大 8 字形的轨迹，图 7.28(b)表示信赖值的权值随学习次数变化的情况，图 7.28(c)表示被生成的识别规则中的一个隶属函数的变化。由图 7.28(d)可知，权值达到与已有的知识同等时，得到了 95%的高识别率。

已有动作指示的修正实验：通过学习把复杂的 8 字形的指示变成了简单的 V 字形的指示。这样，根据简单的 V 字形的指示，识别成了由复杂的 8 字形所给的指示，犬完成了描绘 8 字形的轨迹。图 7.29 表示了修正前和修正后的 8 字形的指示中所用的一个隶属函数的变化。

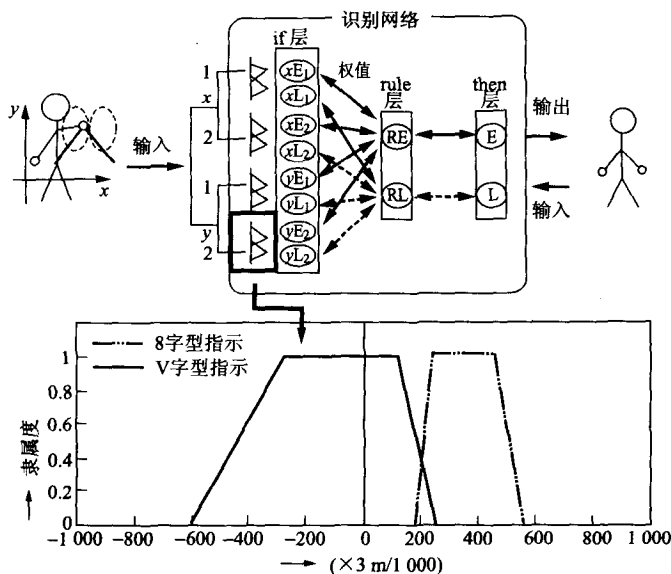


图 7.29 对已有字的指示的修正

(2) GIL法 图 7.30 表示的是 GIL 法的 8 字形和心形的轨迹的精练实验。

首先 把理想的 8 字形的动作的轨迹在纵横向较小设定，然后精练 8 字形的动作的轨迹，进行趋近于理想轨迹的实验。图 7.30(a) 表示了学习前和学习后的轨迹比较。

图 7.31 表示了把理想的心形的动作的轨迹在纵横向较大设定，然后精练心形的动作的轨迹，趋近于理想的轨迹的实验结果。图 7.31(a)表示了学习前和学习后的轨迹比较。

图 7.30(b)和图 7.31(b) 表示了 GIL法进行逐点修正时的理论值与理想轨迹有多大程度的误差。用 GIL 法，能够识别人的意图在性质上的微妙差别，能够对某个动作把数个动作生成规则进行汇集精练。因此，与进行逐个修正相比，能用较少的次数进行学习。

以上，介绍了用宏观信息与人进行对话，为了与人的意图相适合

对图像信号等信息设施上的诸多数值信号进行处理，及基于联想存储的视觉处理的代理机模型。

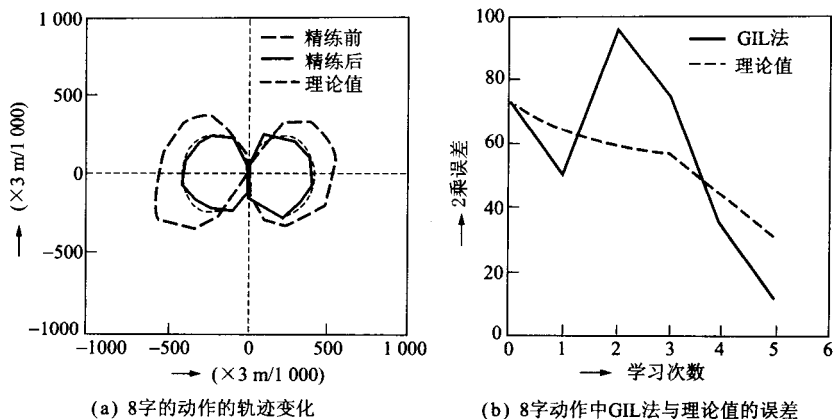


图 7.30 GIL 法的 8 字动作轨迹的精练

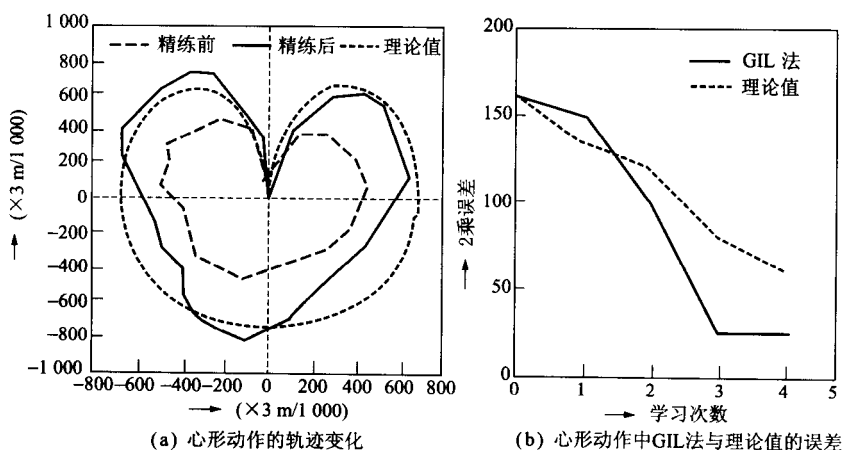


图 7.31 GIL 法的心形动作轨迹的精练

7.3 新型智能信号处理

智能工业系统(IIS)如图 7.11 所示,是由少数的人和多数的代理机构构成的,为了达到大目标,由人来进行粗略判断,然后给代理机下指令,及获取信息的系统。

IIS 中的代理机,必须具备在状况变化而用目前的知识不能对应的情况下,进行知识的学习,构思出新的知识等来适应状况变化的功能。

也就是说,把前节中的实验例子加入到与人的相互作用中,来与外界的相互作用是必要的。与外界的相互作用是指,智能代理机即使遇到外界变化的轻度障碍,也有能够根据当前知识所产生的构思对应状况变化的适应功能,并根据其智能进化功能,有能够回避障碍的功能。

在 5.4 节模糊神经网络的模型圆盘型直升机的在线智能控制实验中,能够实现相当于人的小脑的下位等级的局部循环系统。但是,在实现智能代理机时,就必须用宏观信息与人进行通信,并像人那样构思,自律获得知识。

本节,将介绍这个构思功能,和基于构思功能进行知识进化的多代理机,并说明其并行处理的有效性。

为了达到即使在环境变化时,也能够使用当前知识构思出新的有效知识,自律对应状况变化的目的,要用到图 7.11 所示的基于直觉的代理机模型,即把自下而上处理和自上而下处理统和的代理机模型。

这个代理机模型,有基于混沌力学系统进行微观知识的构思,使用宏观知识对其进行评价的功能。这里,介绍模糊联想存储的基于混沌回想的自组织化方法^[14]。并通过实验说明应用于自律移动机器人的障碍物回避的结果。

而且也应用于使用将同一模型在两台机器人的同时行走中的知识获得。在障碍物回避中,把某个环境中从起点到终点的障碍物回避的控制规则的知识和有关环境的知识模型化,提供给机器人。然后,

在环境变化时，利用有关已有控制规则的知识进行混沌回想，回想候选回避方向。

然后，对各个候选，根据环境模型对到终点的路径进行预测和评价。然后，向评价高的方向运动，并学习其知识。根据以上方法，可以构成适应环境变化的自律移动机器人。在两台机器人同时行走实验中显示了图 7.11 中的通过与人或外界的相互作用，多数代理机进行通信及新的规则的构思。

另一方面，本节中说明的智能阶层模型的阶层间的并行性和直觉联想推理中的回想的并行性，虽然只是一部分，但也可以说是人脑中并行信号处理的模拟。

而且，可以说代理机群中的并行性以及对其扩展形式的进化和淘汰模型是的并行处理的应用，也是自然界中的并行性的一部分的模糊。有关以这样的脑和自然界中的并行性为样板的系统的并行信号处理，作为新的智能信号处理，本书将简单予以介绍。

在下面的 7.3.1 中，将说明为了实现构思功能而使用的模糊联想存储系统和混沌模糊联想存储系统；7.3.2 节将展示基于混沌回想的智能代理机群的自组织化的实验结果；7.3.3 节将说明把 7.3.1 所述的内容应用于使用混沌回想的并行处理和进化淘汰模型的可能性。

7.3.1 模糊联想存储系统中的混沌回想

混沌模糊联想存储系统是在模糊联想存储系统的记忆回想过程中应用混沌回想的系统^[14~16]。由于该系统是模糊系统、神经网络及混沌系统的融合，可以看成是某种软计算的实例。

作为实现在生物体中观测到的混沌的举动研究，有图 7.32 所示的混沌神经网络和混沌模糊联想存储的研究^[6,8,17,18]。图中的混沌模糊联想存储根据模糊标识把这些来自混沌系统的复杂数值信号进行变换，使其成为人能够理解的信号。模糊联想存储中的回想，通常，为图 7.33(a) 所示的稳定回想。

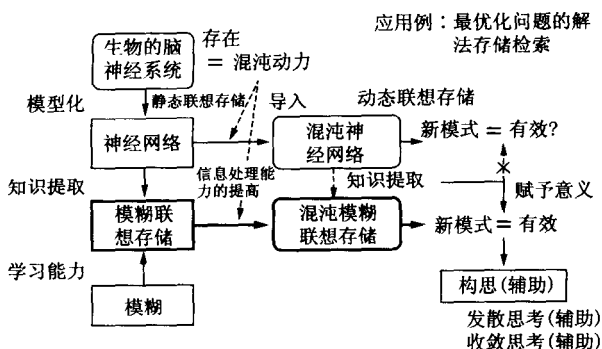


图 7.32 模糊、神经网络、混沌的融合

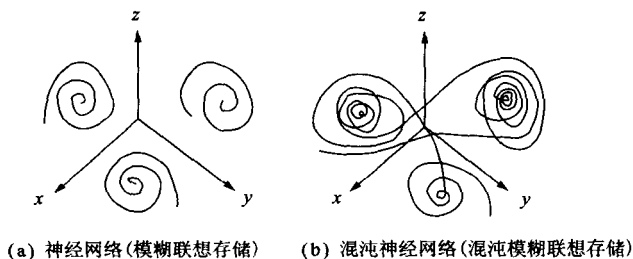


图 7.33 联想存储的动力

但是 通过使用图 (b) 所示的混沌回想方法，动态遍历存储空间，有生成新的存储的可能性。混沌模糊联想存储是实现混沌回想的模糊联想存储，特别是，对在混沌最速下降法（CSD: Chaotic Steepest Descent Method）中使用的回想法进行扩展而应用的。

这个混沌最速下降法是指，对在神经网络的能量曲面中记述运动的力学方程式的散失项，导入如图 7.34 那样的周期性变动的非线性抵抗，混沌性地遍历能量极小解的方法。图 7.35 是能量曲面上的混沌性遍历的概念图。

这样的联想记忆中的混沌性遍历，用式 (7.6)、(7.7) 表示。

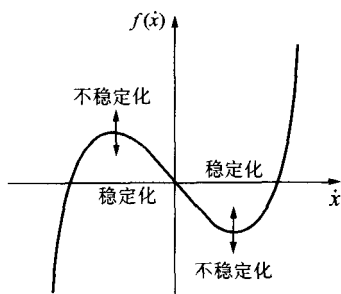


图 7.34 非线性抵抗及其特性变动



图 7.35 能量曲线上的混沌遍历

$$m\ddot{u}_i + f(\dot{u}_i, \omega t) = \epsilon \sum_j w_{ij} a_j + \eta I_i + \begin{cases} 0, & -\beta \leq u_i \leq \beta \\ -\alpha, & u_i > \beta \\ \alpha, & u_i < -\beta \end{cases} \quad (7.6)$$

$$f(\dot{u}_i, \omega t) = [d_0 \sin(\omega t) + d_1] \dot{u}_i + d_2 \dot{u}_i^2 \operatorname{sgn}(\dot{u}_i) \quad (7.7)$$

其中, $\dot{u}_i = du_i/dt$, $\ddot{u}_i = d^2 u_i/dt^2$ 。

u_i, a_i, I_i 分别表示联想存储中第 i 个节点的内部状态, 输出及外部输入, w_{ij} 表示从节点 j 到节点 i 的结合权值, $f(\cdot)$ 表示图 7.34 的非线性抵抗函数, m, ϵ 为正的系数, ω 是为了使非线性抵抗的特性变动的角速度, α 为能量梯度的补正值, β 为节点的内部状态绝对值的范围, η 为对外部输入的权值系数, d_0, d_1 为线性抵抗系数, d_2 为非线性抵抗系数, $\operatorname{sgn}(\cdot)$ 表示符号函数。还有, 式(7.6)的外部输入项 ηI_i 是新增加的项^[14]。

图 7.36 表示的是设定某个外部输入，使与混沌发生的非线性抵抗有关的系数 d_2 变化时的网络状态转移。

图中的 x 轴为外部输入和内部状态的欧几里德距离， y 轴为内部状态的能量的差分。还有，与用 $1/d_2$ 表示的非线性的强度为弱，中，强时对应的混沌的回想的结果分别表示在图 7.36(a),(b),(c) 中。

从这些结果可以看出，通过增强非线性，混沌的回想范围在扩大。而且，这时，能量的变化达到趋近于零的新的暂时稳定点，即新的模式的构思。

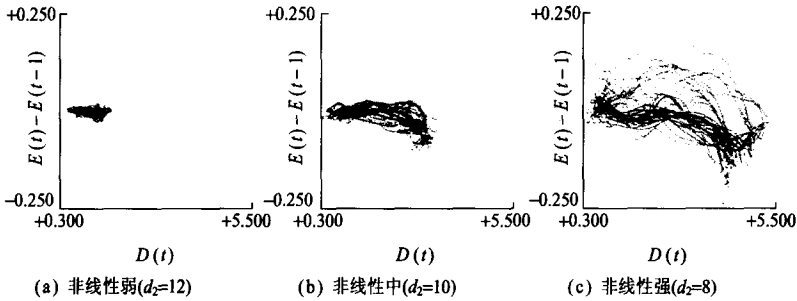


图 7.36 非线性的强度与回想范围
(© 1995FAM Project, LIFE)

这样的混沌模糊联想存储系统有以下两个功能。

功能 1：在存储的模式中，对在输入模式的近范围内存在的存储模式进行动态回想；

功能 2：对没让存储但在意义上是有效的，新的模式进行回想。

这里 功能 1 在图 7.33 中已经说明。功能 2 在混沌性存储的探索中，没让存储的模式多数被回想，揭示了其中有含有意义上有效模式的可能性。

这是因为，混沌模糊联想存储系统中，通过在网络的各节点中保持用模糊标识表示的意思，由于明确地表示出知识，可以判断被回想的模式是否有效，作为新的模式有使用的可能性。

根据以上的功能，混沌模糊联想存储系统具有使用在联想存储上存储的已有知识能够生成新的知识的所谓构思功能的特征。因此，用混沌模糊联想存储系统来构思(辅助)是可以实现的。图 7.37 表示了存储空间中的混沌的回想和构思功能。图中表示了从初期模式 X 开始的构思过程。没有非线性(的情况下，与通常的联想存储相同，回想最近的已存储模式 A。弱非线性条件下，回想其周围的存储模式 A、B。随着非线性增强，可以得到几个认为有效的新模式。这个过程，可以作为发散性思考^[19]的一个方法来辅助使用。

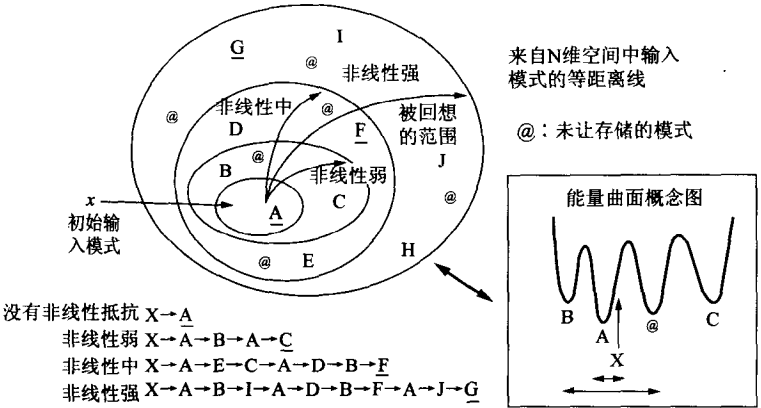


图 7.37 存储空间中的混沌回想与构思功能

把由发散性思考得到的有效辅助模式之一作为初始条件给出，把非线性条件减弱来探索其周围的话，就能够得到更有效的模式。这个过程可以说是收敛思考的一个辅助方法。

7.3.2 基于混沌回想的构思功能和自组织化

用自律移动机器人的障碍物回避的实验例子来说明前述的构思能力的利用方法。

1. 自律移动机器人系统

机器人系统 如图 7.38 所示，由跟踪、传感器信息计算和机器人控制等部分构成。

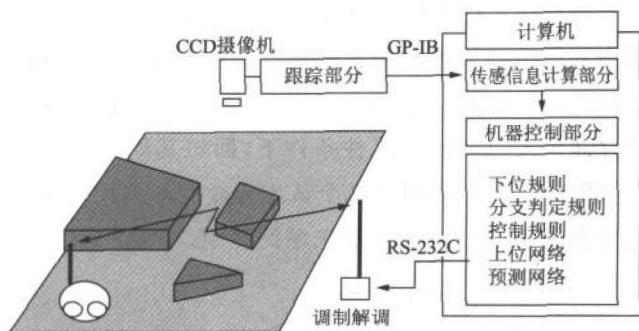


图 7.38 移动机器人的障碍物回避系统

跟踪部分由 CCD 摄像机，色彩跟踪器，色彩提取器组成，使用颜色信息检出机器人的位置，以 1/60 每秒得到二维坐标系中的坐标值。传感信息计算部分 如图 7.39 所示，作为机器人的眼睛由摄像机取像，并根据被推定的图像信息计算障碍物的面积值。

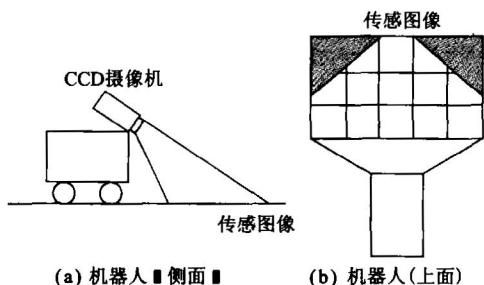


图 7.39 传感图像

另外，将其值作为机器人到达终点的粗略距离和方向，用模糊标识的定性表示进行变换。在机器人控制部分中，由传感信息计算部分输出的图像和从机器人到终点的粗略距离和方向，决定下面要行进的方向，然后机器人向决定的方向移动。图 7.38 右下方表示的机器人

控制部分中的层次性规则，是前面说明过的 Rasmussen 的智能系统模型的扩展。以下是本知识的说明和行进方向的确定过程。

下位规则：有使用传感图像来判定是否为分支的分支判定规则和决定要行进方向的控制规则。控制规则对传感图像的开辟方向的重心进行计算。分支判定规则，在道路已知有 2 条时，突然眼前开阔时，视线被障碍物遮挡时，分别判定为分支。

上位网络：由于要在分支中决定行进的方向，这个知识用前面说过的联想存储网络构成。而且，如图 7.40，通过对混沌模糊联想存储系统的扩展，可以对应需要进行构思处理。

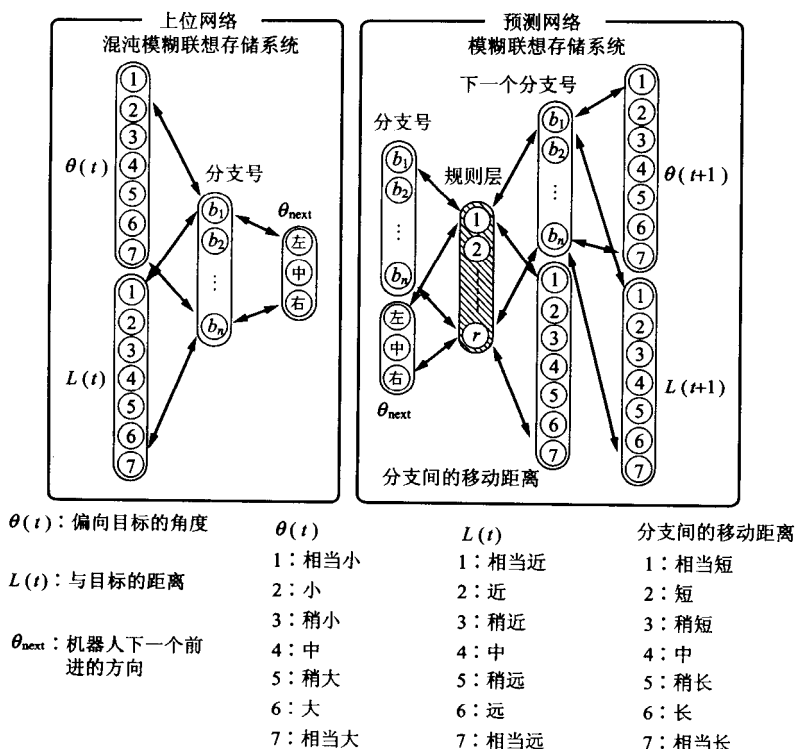


图 7.40 上位网络与预测网络

预测网络：这个知识，是把外部环境在模糊联想存储中存储，如图 7.40 所示，在预测下一个分支中使用。

机器人控制部分的流程，如图 7.41 所示，首先用分支判定规则判定是否为分支。

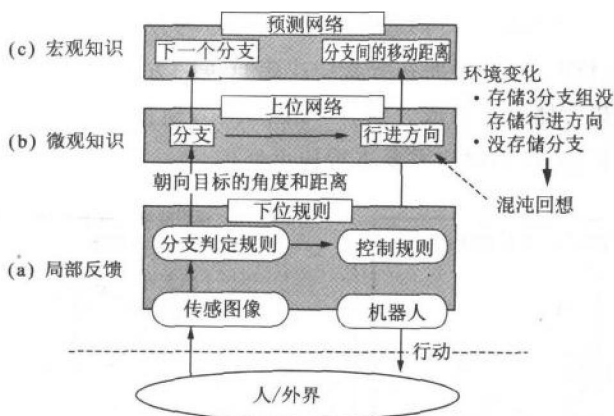


图 7.41 机器人控制部

不是分支时，按照下位控制规则，向没有障碍物的方向行进。是分支时，是否存储其分支，用上位网络考察。作为分支，如果没有存储，判断为环境变化，然后停止。若存储了，用预测网络比较是否与预测的分支相同。

如果与被预测的分支不同，判断为环境变化了然后停止。如果与被预测的分支相同，在其分支是否存储了移动方向，用上位网络考察。如果存储了就按其方向行进。如果没有存储就地停止。用这个层次知识结构进行了以下实验。

2. 路径计划实验

机器人控制部分中，当判断为环境已变化时，在上位网络进行混沌回想来回想回避方向。当到达和预测的分支不同的已知分支时，在

这个地点进行混沌回想。

还有，在和预测的分支不同的新的分支发生时，返回到前面的分支进行混沌回想。图 7.42 是这个分支中的构思仿真的混沌回想过程。

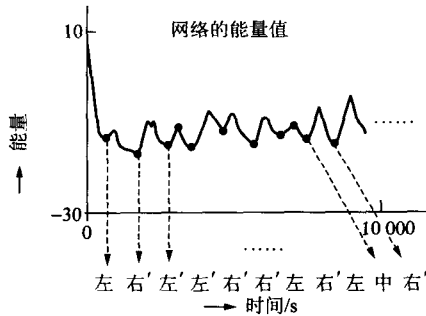


图 7.42 能量极小时的输出 (反转模式)

在这个构思中，与图 7.36 所示的相同，由于与非线性相对应在一定范围内进行遍历，有对极端远的地方不能被回想的性质。

下面说明应用混沌模糊联想存储系统的路径计划的方法。

(1) 路径计划算法

把变化的状况编入预测网络；

用上位网络进行混沌回想，对回避方向的候选进行回想；

把由步骤 得到的方向输入给预测网络，预测下一个分支；

由步骤 被预测的下一个分支中向接近终点的方向行进，并使用预测网络对再下一个分支进行预测；

继续执行步骤 直到终点被预测，这里，把各分支间机器人行进的距离和离终点的到达度作为评价值进行计算；

⑥ 对在步骤 中被回想的方向的候选执行步骤 到 ；

⑦ 在由步骤 ~⑥得到的到终点的路径中，选择评价值最高的路径；

⑧ 当前的分支中向与步骤 ⑦的路径对应的方向行进时，用预测

网络预测下一个分支；

⑨ 如果步骤⑧中预测到达了终点，则执行步骤⑩，如果没有到达，重复执行步骤⑧；

⑩ 最后剩下的路径为应该行进的路径，于是，机器人就沿这条路径的方向行进；

⑪ 把与实际变动的路径相对应的规则作为控制规则，用其对上位网络进行精练；

⑫ 在出发地重复步骤⑧~⑪。

本算法中，步骤⑧能够进行预测网络的修正，如图 7.40 所示，是因为明确地表示了知识。步骤⑪把回避方向进行种种回想，是用了发散思考的辅助方法。

还有，通过用步骤⑦对评价高的进行选择，路径的探索范围可以缩小。这是因为用了收敛思考的辅助方法。因此，本算法可以实现具有发散思考和收敛思考两方面的构思。

还有，由于使用构思的结果自律对知识进行精练，本算法具有自组织化功能。而且，接着步骤⑪执行⑫能够探索环境变化后的从始点到终点的路径中评价价值最高的路径。

(2) 实验结果用图 7.43 所示的环境（障碍物、始点、终点）按照前述的路径计划算法进行了实验。

这里，环境在纵横方面都取为（1000，1000），机器人的大小取为纵 150×横 100，机器人的视角取为纵 300×横 500（分割成 3×5 块）。

还有，也对上位控制知识，存储了图示的路径。

在这样的不变环境状况下，通过使用下位的控制规则和上位网络，能够到达终点。因此，障碍物的变化，考虑下面两种状况。

状况 1 如图 7.44 所示，增加障碍物。分支 1 之后应该走向分支 2，但却到达了与预测的分支不同的分支 3。这时，在分支 3 进行混沌回想。

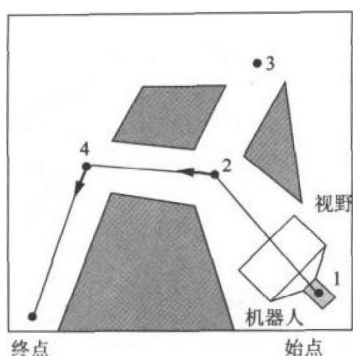


图 7.43 外部环境

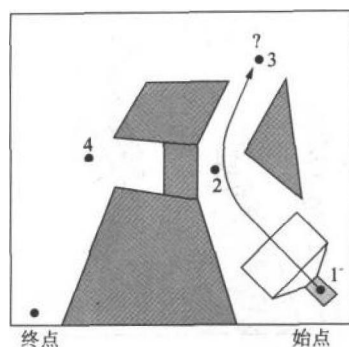


图 7.44 外部环境变化 1

状况 2 如图 7.45 所示，增加障碍物。分支 2 之后应该走向分支 4 但却到达了新的分支 5 这时 返回到分支 2 进行混沌回想。

在这样的条件下，计算机仿真的机器人的轨迹见图 7.46 和图 7.47。如这些图所示，用本算法是可以自律规划与环境变化对应的路径的。

还有，如果在始点就进行混沌回想，可以发现图 7.48 所示的路径。

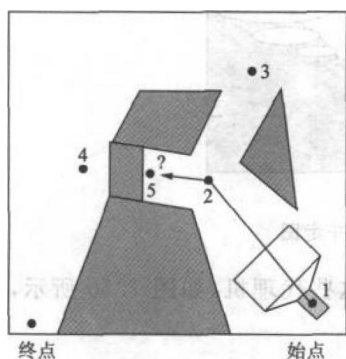
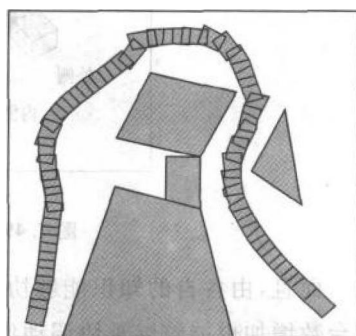


图 7.45 外部环境变化 2



阻 7.46 实验结果 1

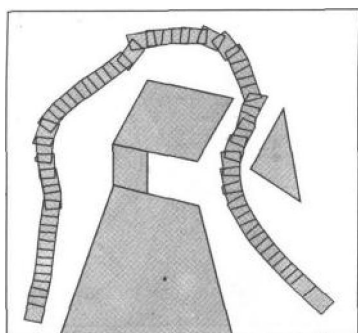


图 7.47 实验结果 2

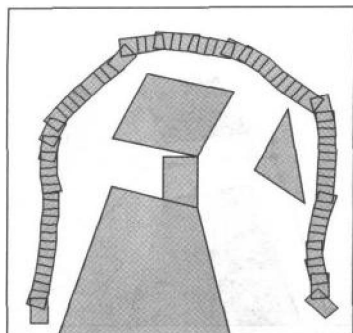


图 7.48 实验结果

3. 两台并走实验

对人来说，两个人并行行走时，通常相互协调步伐行走。在转弯时外侧的人稍快，内侧的人稍慢。在图 7.49 的机器人代理机中就必须根据各自的知识进行调整，进行动作的协调。

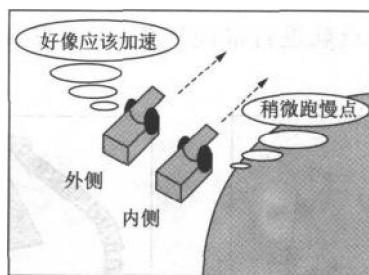


图 7.49 两台并走图

而且，由各自的知识能够协调的这些代理机，如图 7.50 所示，即使台数增加也能够继承协调动作。

图 7.50 中，显示了按人的指示组成横排和围起来状形的样子。这样的与人的协调动作和形式动作，就是机器人帮助人行走，帮助人从床上起来的必须有的基础性功能。

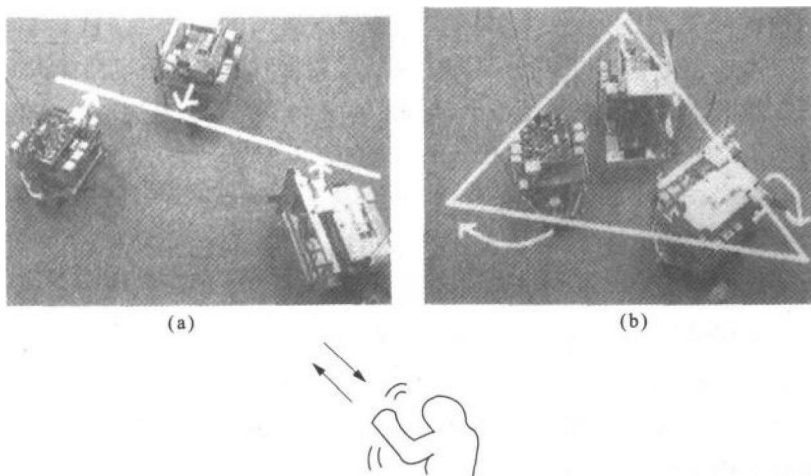


图 7.50 与人协调的代理机器人群

用与图 7.41 的路径计划机器人的结构相同的 3 层知识模型，来构成多代理机机器人的知识。图 7.51 表示了这个机器人控制部的知识结构。

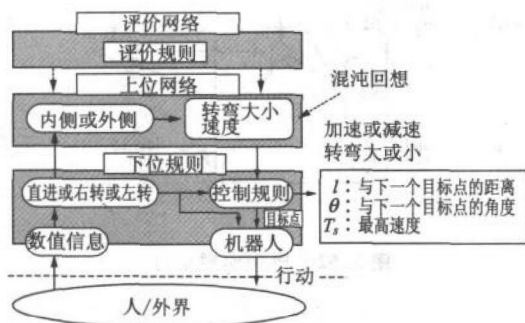


图 7.51 两台并走中的机器人控制部

这里，机器人一边把握对方的动作和与之相对的自己的状态，一边行动。为了使协调动作很好地进行，根据混沌回想来构思上位网络

的知识。

图 7.51 中，上位网络具有为了与状况变化对应的使用模糊标识的定性知识，而下位层的数值输出，与此依存进行变化。就是说，对上位网络和下位的规则，用图 7.52 所示的概念集合（CFS）^[20] 进行构成。

图 7.51 中，下方是下位规则，上方是上位网络。在下位规则中，在左侧进行数值输入，在右侧进行数值输出。数值输入被变换成“向右变大”和“向左变小”等的标识和真理值。

对于输出，输出速度和转向值。作为上位网络中的定性知识，有“是否单独移动”、“是否两台并行行走”；如果是两台并行行走，对下一个转弯方向，自己成为内侧还是成为外侧”，并根据这些信息，必须使输出值与状况相对应变化。

这个知识结构，与前述的路径计划机器人的结构相同，可以说是对很多智能系统有效的方法。

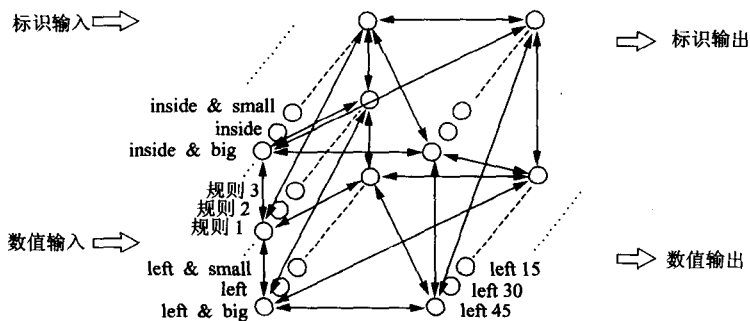


图 7.52 概念模糊集合

7.3.3 并行处理以及进化和淘汰模型

把前述的实现两台并行行走的代理机机器人看作与人同等的个体群的话，自然就是按个体的并行处理。根据代理机的构思处理，具有

新知识的新的代理机的生成，与环境的变化相对应不断重复进行进化和淘汰的并行处理的自律模型，从现实世界的观点看，可以说是自然模型。图 7.53 显示了把进化和淘汰模型在工学上进行模拟向创发系统的扩展。图中表示了代理机智能的进化，也就是说，在这里展现了一个信号处理算法的自律进化模型的例子。由个体组成的代理机机器人群，在个

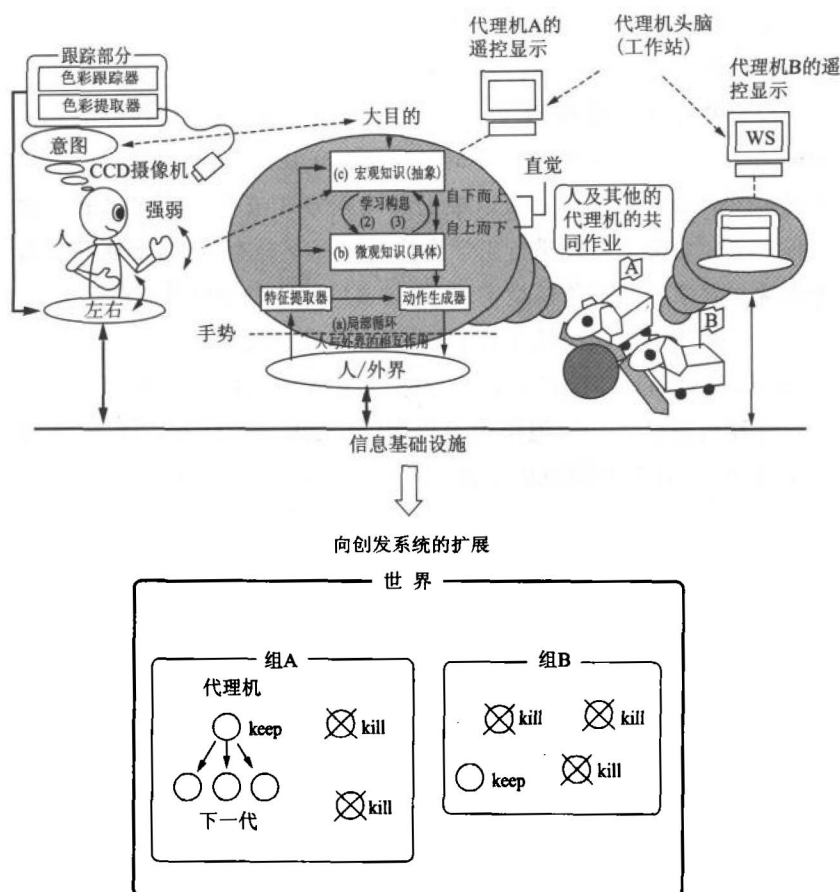


图 7.53 向创发系统的扩展及并行处理

别环境状况下根据构思生成进行新的信号处理的子孙。在子孙代理机中，除了在个别环境状况中能够继续生存的代理机外被淘汰。这样，被并行处理的智能的进化模型就可以在工学上进行模拟。

这里，与个体对应的代理机是指，对信号自律进行处理，而且是自律发出信号的个体模型。一个个体内的器官，例如可以说相当于脑的信号处理的并行化。

在本节中，将说明在工学上这些信号处理的并行化和代理机的自律动作的并行化，方法及其实现的可能性。

(1) 并行计算机 **A-NET** 以并行目标指向的概念为核心的总体结构 A-NET (actors network) 的开发研究正在进行。目标指向是指，把现实世界的对象进行抽象以数据和编程方式进行一体化来表示的编程方法。对生态系统进行工学模拟的情况下是有效的，个体和器官等各构成要素用其抽象模型实现，通过通信的收发可以执行程序。

本计算机中也实现其并行动作。A-NET 作为并行目标指向语言的 A-NET 和编程辅助环境，把 1000 台规模的 MIMD 分散存储器型并行计算机进行统一处理作为前提，其研究正在进行着。图 7.54 表示了这个 A-NET 计算机的结构。

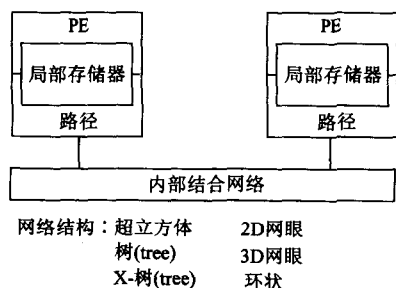


图 7.54 A-NET 计算机的结构

作为这个计算机的处理要素的节点处理器，由要素处理器(PE)和结构存储器，通信路径选择的专用 Route 组成，网络结构，除了有超立

方体等标准的 6 种外，还可以定义使用任意结构。

还有，把目标最优分配给多种结构的 Allocate 的综合也是其显著特征。使用的编程语言 A-NET 是基于 actor 理论的语言，是在 A-NET 计算机的要素处理器上工作的描述语言。

A-NET 用目标单位描述程序。在目标中可以定义数个方法，对目标通过发送信息，能够启动任意方法。

(2) 并行处理的仿真图 7.55 是在 A-NET 计算机上进行两台并行行走仿真情况下的机器人轨迹。

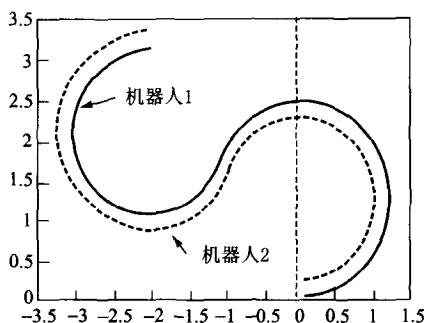


图 7.55 两台并走仿真

图 7.56 是并行型程序中，把各目标分配给节点处理器的工作状态图，横轴表示时间，纵轴表示节点处理器的顺序号。这些处理，与各代理机脑中的信号处理的并行化相对应。

在节点处理器顺序号的旁边，把在那儿被分配的目标用括号表示，把各目标进行分配给第 0 号分配 then 层，第 1 号分配规则层，第 2 号分配 main 层，第 3 号分配 if 层。这样，就实行了代理机中的信号处理，即联想推理的并行化。各层与联想存储器的层相对应。逐次型程序和并行型程序的实行时间的比较表示在图 7.57 中。

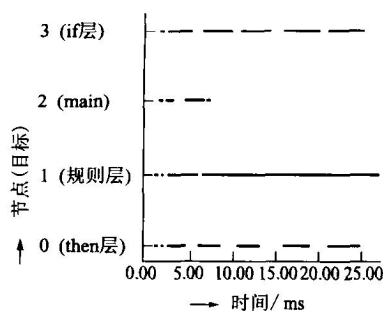


图 7.56 并行联想处理的实行图

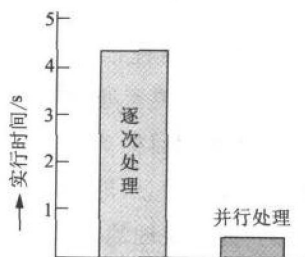


图 7.57 逐次处理与并行处理的实行时间比较

总的实行时间，用逐次型为 4.33s 而用并行型为 0.39s。所以并行型的处理速度约为逐次型的 11 倍。现实个体中的信号处理，比如，人体中的信号处理也是并行处理，可以说，这个例子中也用了自然方法来实现处理的快速化。

而且，与个体对应的代理机的自律信号处理也可以说是自然的。对多代理机的自律进化和淘汰模型进行仿真的研发系统中，并行处理也是自然的，能够实现并行处理的高速化。

图 7.56 所示的仿真中，也是代理机分别并行进行仿真，而且，按照右和左等的各个状况进行了并行性仿真。通过这个并行处理实现了两台协调动作，获得了相当于图 7.52 中的上位网络的定性模糊规则。

从以上结果可以看出，神经网络和模糊信号处理能够用自然形式模拟人等脑中的并行化信号处理，能够快速，实时地进行信号处理。而且，在实现自律代理机的进化的创发系统中，也能够模拟生态系统的自然并行模型。

附录

模糊系统与稳定性

- A.1 线性系统中有理函数的不可约分解表示
- A.2 基于等价多重线性系统的模糊系统的公式化
- A.3 实现基于在线稳定判别条件导出的学习控制

我们要叙述的是，基于由状况和操作模型（控制器）构成的模糊模型式的语言表示抽出模糊规则，通过学习精练其知识，导出模糊系统的德定判别条件的方法。因此，用等价多重线性系统把本书图 5.10 的模糊系统会式化，并根据线性系统理论说明得到用于稳定化控制器的条件的方法。用这个方法说明 5.4.3 节的实验所说明的圆盘型直升机的悬停学习控制系统的稳定条件。

A.1 线性系统中有理函数的不可约分解表示

作为数学准备，先说明一下线性系统中有理函数的不可约分解。

A.1.1 符号定义

R_p, S, U 定义如下：

R_p ：固有有理数行列式的集合；

S ：稳定时固有有理数行列式的集合；

U 单位模($G \in S$ 且 $G^{-1} \in S$)有理传递函数行列式的集合。

A.1.2 不可约分解表示

把控制对象 $P_i \in R_p$ 作为多不可约分解，就能够进行稳定的有理函数的分解表示，用下式表示：

$$P_i = N_i D_i^{-1} = \tilde{D}_i^{-1} \tilde{N}_i \quad (\text{a. 1})$$

$$\begin{bmatrix} Y_i & X_i \\ -\tilde{N}_i & \tilde{D}_i \end{bmatrix} \begin{bmatrix} D_i & -\tilde{X}_i \\ N_i & \tilde{Y}_i \end{bmatrix} = I \quad (\text{a. 2})$$

式中的 $N_i, D_i, Y_i, X_i \in S$ 为右不可约， $N_i, D_i, Y_i, X_i \in S$ 为左不可

约。式(a.2)为多不可约分解中的 Bezout 等式,有 $Y_i, X_i, Y_i, X_i \in S$ 。

A.1.3 内部稳定性

图 a.1 的控制系统表示为

$$\begin{bmatrix} I & P \\ -C & I \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (\text{a.3})$$

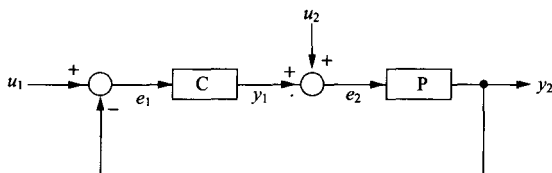


图 a.1 控制系统与内部稳定性

这里假定控制系统为 well-posed,

$$\begin{bmatrix} I & P \\ -C & I \end{bmatrix}^{-1} = \begin{bmatrix} (I+PC)^{-1} & -P(I+CP)^{-1} \\ C(I+CP)^{-1} & (I+CP)^{-1} \end{bmatrix} \in S \quad (\text{a.4})$$

\Leftrightarrow 若 u_1, u_2 为有界输入, 则闭环内所有信号 (e_1, e_2, y_1, y_2) 有界。

根据用式(a.1)和式(a.2)不可约分解表示的 Youla 参数化, 图示控制系统进行内部稳定化的补偿器 C 的族(集合 $S(P)$)为:

$$S(P) = (Y - R_1 \tilde{N})^{-1} (X + R_1 \tilde{D}), R_1 \in S \quad (\text{a.5})$$

$$S(P) = (\tilde{X} + DR_2)(\tilde{Y} - NR_2)^{-1}, R_2 \in S \quad (\text{a.6})$$

A.1.4 强稳定化的可能性

对于 $P \in R_p$, 补偿器 $C \in S(P) \cap S$ 存在时, P 为强稳定化是可能的。

P 为强稳定化可能性 $\Leftrightarrow P$ 满足 p. i. p. (parity interlacing prop-

erty) 条件^[3]。

另外，强稳定化可能性是指，用稳定的补偿器能够对闭环系统进行内部稳定化。

A. 1.5 小增益定理

图 a. 2 中取 $P \in S, C \in S$ 。这时

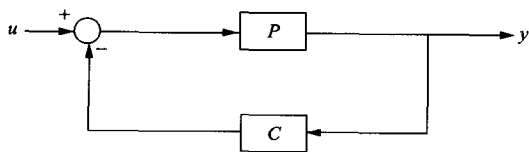


图 a. 2 控制系统与小增益定理

$\|PC\|_{\infty} < 1 \Rightarrow$ 图示闭环系统为内部稳定

式中, $\|\cdot\|_{\infty}$ 表示 H_{∞} 范数。 H_{∞} 范数是指在图 a. 3 的奈奎斯特图线中所表示的最大增益。即表示为

$$\|P\|_{\infty} = \sup\{|P(j\omega)|, -\infty \leq \omega \leq \infty\} \quad (\text{a. 7})$$

因此，若用奈奎斯特图线表示小增益定理，如图 a. 4 所示， $\|PC\|$ 在半径为 1 的圆内。

有了不可约分解表示的数学准备，就可以进行模糊系统的公式化。

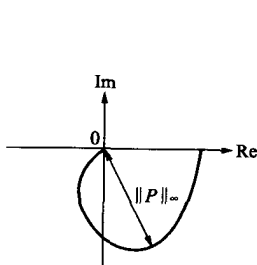


图 a. 3 控制对象与 H_{∞} 范数

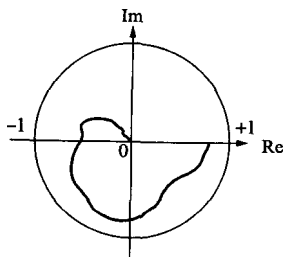


图 a. 4 小增益定理

4.2 基于等价多重线性系统的模糊系统的公式化

讨论本模糊系统是否对控制对象进行了稳定化。因此，把本书图 5.10 的模糊系统作为等价多重线性系统进行公式化。

A.2.1 同化多重线性系统和模糊控制器

基于模糊模型的模糊控制器，用下面所示形式的模糊规则来实现。

把对应于控制对象变动的控制知识，用“if[（表示代表变动状况的参照模型）模糊条件命题]，then[（与代表状况的参照模型对应的操作模型的）输入输出函数]”的模糊规则的知识表示来实现控制器。用 if 部的模糊条件命题把参照模型进行区分并赋以特征。在 then 部的输入输出函数中记述与其参照模型对应的操作模型。这个结构表示在正文的图 5.10。

图中 μP_i 为对 P_i 的隶属函数。 μP_i 是对 P_i 附加特征的模糊集合，是把控制对象的特性变动向模拟空间的映射。在这个空间，由于控制对象的特性变动转换成观测特征量而成为受影响程度的测度，就把这个量称为特性类似度。观测到的控制对象的特征量被映射到模拟特性变动的空间。

直升机控制规则中，与 if 部的高度 h 对应的隶属函数来评价特性类似度。把控制对象 P_i 向模拟变动的空间映射的模糊集合取为 μP_i 。这里考虑以三个特征参照模型 P_1, P_2, P_3 代表参数缓慢变动的 P_i 为例。当 P_i 在时刻 t 的线性近似模型为 P_t ，用 $[A, B, C, D]$ 表示考虑变动的扩大系统的状态空间表示的矩阵时， P_t 可以近似假定为下式：

$$\begin{aligned} P_t &= [A, B, C, D] \\ &= \sum_{i=1}^3 \alpha_i [A_i, B_i, C_i, D_i] \end{aligned} \quad (\text{a. 8})$$

式中， $\alpha_1 + \alpha_2 + \alpha_3 = 1, \alpha_i \geq 0 (i=1, 2, 3), [A_i, B_i, C_i, D_i]$ 为参照模型 P_i

的状态空间表示的矩阵。式(a.8)表示了控制对象的参数合成。模糊控制器对特征量进行观察,并根据使用了与各参照模型 P_i 的特征类似度的模糊推理,来决定操作模型 C_i 的贡献度 γ_i 。控制器的操作量为 u^* 下式:

$$u^* = \sum_{i=1}^3 \gamma_i \cdot C_i(y) \quad (\text{a.9})$$

式中, $\gamma_1 + \gamma_2 + \gamma_3 = 1, \gamma_i \geq 0 (i=1,2,3)$, 操作模型 C_i 为与 P_i 相对应设计的补偿器, y 是观测到的控制量。

式(a.9)的模糊控制器有下述两种形式:

补偿器 C_i 没有个别状态的情况下,就成为调整补偿器参数的单一适应控制器。这时,用式(a.9)的 $\gamma_i (i=1,2,3)$ 合成的补偿器取为 C^* ;

补偿器 C_i 内部有个别状态(有个别观察等)的情况下,就成为数个补偿器的合成。

这种表示在多重线性系统下可以近似成等价。下面叙述为谋求等价多重线性系统的公式化,利用基于在线性系统中使用的稳定有理函数行列式的分解表示的方法,求这种稳定化补偿器的类的方针。

使用单一适应控制器 C^* 时,补偿器 C_i 满足下面的[条件1与条件2],或者[条件1与条件3]为充分条件。

这里取 P_i 为参照模型的传递函数,取 C_i 为补偿器的传递函数。还有取 C^* 为模糊控制器的传递函数,取模糊控制器的假定控制对象的传递函数为 P^* 。

条件1: C_i 为 P_i 的控制器。根据 uP_i 的定义, C_i 为对与 P_i 相近的特性类似度的控制对象全部进行稳定化的补偿器。

条件2: C^* 在稳定 P^* 的同时,也稳定 P_i 。换言之, C^* 同时稳定 P^* 和 P_i 。

而且作为条件2的必要条件,得到条件3。

条件3: C_i 在稳定 P_i 的同时,对特性类似度附近的别的参照模型

P_j (以及 μP_i 内的控制对象) 进行同时稳定化。

条件 1 为普通补偿器的条件。

良好的模糊推理, 由于 P^* 和 P_i 的特性类似度十分相近, 所以条件 1 包含条件 2。条件 2 和条件 3 作为“同时稳定化问题”被公式化。而且, 在用数个个别补偿器的情况下, 要增加下面的条件 4 予以制约。

条件 4 补偿器 C_i 和 C_j 的参照模型 P_i , 还有与别的特性类似度附近的参照模型 P_j 对应的补偿器 C_j 的凸结合也对 P 进行稳定化。

我们把条件 4 作为“凸结合型多重补偿器问题”进行公式化。这里把基于模糊模型的模糊控制器, 作为“同时稳定化问题”和“凸结合型多重补偿器问题”进行公式化。利用控制理论中的稳定有理函数行列式的分解表示的方法, 来求稳定化补偿器的类。作为第一步, 把问题简化分为下面两个问题, 来明确补偿器的类。

同时稳定化问题把控制对象用数个线性模型表示, 求把这些模型同时稳定化的补偿器的类这样的问题。

凸结合型多重补偿器问题求对于一个控制对象使数个补偿器进行凸结合控制时的, 能够稳定化的补偿器的问题。

A. 2. 2 同时稳定化问题

这里求控制对象为 P_1, P_2 两个时的稳定化补偿器 C_1 的类。

本书中图 5.11 表示了这个同时稳定化问题。由于存在与 P_1 的特性类似度相近的 P_2 根据变动从 P_1 向 P_2 从 P_2 向 P_1 切换。用多不可约分解表示控制对象 $P_1, P_2, \in S$ 。按式 (A. 1) 式 (A. 2) 说明的那样, 可以得到下面稳定有理函数的分解表示:

$$P_i = N_i D_i^{-1} = \tilde{D}_i^{-1} \tilde{N}_i, \quad i=1, 2 \quad (\text{a. 10})$$

$$\begin{bmatrix} Y_i & X_i \\ -\tilde{N}_i & \tilde{D}_i \end{bmatrix} \begin{bmatrix} D_i & -\tilde{X}_i \\ N_i & \tilde{Y}_i \end{bmatrix} = I, \quad i=1, 2 \quad (\text{a. 11})$$

式中 式 (a. 10) 的 $N_i, D_i, Y_i, X_i \in S$ 为右不可约, $N_i, D_i, Y_i, X_i \in$

S 为左不可约。而且, 式(a. 11)为多不可约分解中 Bezout 等式 并有

$$Y_i, X_i, \tilde{Y}_i, \tilde{X}_i \in S。$$

这里 根据式(a. 5)的 Youla 参数化法 对 P_i 的补偿器 C_i 的类为

$$C_i = (Y_i - R_i \tilde{N}_i)^{-1} (X_i + R_i \tilde{D}_i) \quad (\text{a. 12})$$

而且, 根据

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} Y_1 & X_1 \\ -\tilde{N}_1 & \tilde{D}_1 \end{bmatrix} \begin{bmatrix} D_2 \\ N_2 \end{bmatrix} \quad (\text{a. 13})$$

若定义 $A, B \in S$ 则可得对 P_1, P_2 同时稳定化的必要充分条件。

定理 1(同时稳定化的必要充分条件) 由式(a. 12)的控制器 C_1 的 Youla 参数化法, 同时稳定化的必要充分条件满足

$$\begin{aligned} C_1 &= (Y_1 - R_1 \tilde{N}_1)^{-1} (X_1 + R_1 \tilde{D}_1) \\ &= (Y_2 - R_2 \tilde{N}_2)^{-1} (X_2 + R_2 \tilde{D}_2) \end{aligned} \quad (\text{a. 14})$$

的 $R_1, R_2 \in S$ 存在。而且这个条件与满足

$$(A + RB) \in U \quad (\text{a. 15})$$

的 $R \in S$ 存在等价。式中的 A, B 为右不可约。

关于这些证明予以省略。式(a. 14)及式(a. 15), 表示了变动控制对象间的同时稳定化的可能性, 这个条件成为根据模糊模型构成模糊控制器方面的重要条件。

A. 2. 3 凸结合型多重补偿器问题

模糊控制器在合成个别数个控制器时, 为了稳定性要增加制约。对于一个 P_1 想到利用二个补偿器 C_1, C_2 的凸结合来求稳定化补偿器的类。这个条件成为凸结合型多重补偿器的制约。

本书的图 5. 12 中表示了凸结合型多重补偿器问题的构成。观测控制对象的参数变动的情况下, 用模糊推理进行 C_1, C_2 的凸结合, 适应调整控制器, 调整的操作量为

$$\begin{aligned} u^* &= \gamma_1 \cdot C_1 \cdot y + \gamma_2 \cdot C_2 \cdot y \\ &= \{\gamma_1 \cdot C_1 + (1-\gamma_1)C_2\}y \end{aligned} \quad (\text{a. 16})$$

式中, $\gamma_1 + \gamma_2 = 1, \gamma_1 \geq 0, \gamma_2 \geq 0$ 。

由式(a. 16), 补偿器 C 定义为

$$C = \gamma_1 \cdot C_1 + (1-\gamma_1)C_2 \quad (\text{a. 17})$$

这里, 假定 P_1 为强稳定化, 其稳定的稳定化补偿器之一为 C_1 。由 P_1 的多不可约分解, 满足

$$P_1 = N_1 D_1^{-1} = \tilde{D}_1^{-1} \tilde{N}_1 \quad (\text{a. 18})$$

$$\begin{bmatrix} I & C_1 \\ -\tilde{N}_1 & \tilde{D}_1 \end{bmatrix} \begin{bmatrix} D_1 & -C_1 \\ N_1 & I \end{bmatrix} = I \quad (\text{a. 19})$$

式中的 $N_1, D_1 \in S$ 为右不可约, $\tilde{N}_1, \tilde{D}_1 \in S$ 为左不可约。

这里若用 Youla 参数化法表示对 P_1 进行稳定化的 C_2 有

$$C_2 = (C_1 + D_1 R)(I - N_1 R)^{-1} \quad (\text{a. 20})$$

式中 $R \in S$ 为自由参数。补偿器 C 对 P_1 进行稳定化的充分必要条件为

定理 2 (凸结合稳定化补偿器的充分必要条件) R 为 $-\gamma_1 N_1$ 的稳定化补偿器 凸结合补偿器 C 为对 P_1 进行稳定化的充分必要条件。

【证明】 将定义凸结合补偿器 C 的式(a. 17)变形可得下式:

$$C = (C_1 + D_1 R - \gamma_1 R)(I - N_1 R)^{-1} \quad (\text{a. 21})$$

对于任意的 $\gamma_1 \in [0, 1], C$ 为对 P_1 进行稳定化的充分必要条件为, 对于各个 γ_1 满足下式的 $Q \in S$ 存在。

$$C = (I - Q \tilde{N}_1)^{-1} (C_1 + Q \tilde{D}_1) \quad (\text{a. 22})$$

由式(a. 21)和式(a. 22)可得下式

$$Q(I - \gamma_1 N_1 R) = (1 - \gamma_1)R \quad (\text{a. 23})$$

这里, C 对 P_1 进行稳定化的条件为, 满足 $Q \in S$ 的 Q 存在。

这个条件 根据式(a. 23) 变为 $(I - \gamma_1 N_1 R) \in U$ 。而且 这个条件

与 R 对 $-\gamma_1 N_1$ 进行稳定化的条件等价。

[证毕]

特别地 由 $\|\gamma_1 \tilde{N}_1 R\|_\infty \leq \|\tilde{N}_1 R\|_\infty$ 的关系 ($\|\cdot\|_\infty$ 为 H_∞ 范数), 应用小增益定理可得下面的充分条件。

定理 3 (凸结合稳定化补偿器的充分条件) 如果 $\|N_1 R\|_\infty < 1$, 对于任意的 $\gamma_1 \in [0, 1]$ 补偿器 C 对 P_1 进行稳定化。

A.3 实现基于在线稳定判别条件导出的学习控制

把上述讨论的稳定性结果应用到第 5 章的直升机学习控制。

A.3.1 直升机的模糊模型

根据操纵者对直升机动态特性的知识, 来构成高空飞行和低空飞行等不同状况中的表示参数变动的模型。参考一般直升机的框图确定模型, 然后, 进行一问一答式的各种鉴定实验, 来逼近直升机的特性。直升机的近似特性为 4 次传递函数 P_v :

$$P_v = \left[\frac{ke_0}{s(s^2 + kd_3s + kd_2)} \frac{ke_0}{s^2(s^2 + kd_3s + kd_2)} \right]^T \quad (\text{a. 24})$$

这里 k 为变动参数 ($k_3 \leq k \leq k_2$), $e_0, d_i, k_i, (i=2, 3)$ 为正系数。低空飞行中由于来自陆上的地面效果作用, 操纵的灵敏度变高 ($k=k_2$ 的情况)

还有, 由于地面的湍流, 干扰将增加。高空飞行中地面效果减小, 操纵的灵敏度降低 ($k=k_3$ 的状况)。式 (a. 24) 的参数值表示在表 5.2。操纵者可以利用直升机具有的二个特征模型 (参照模型)。为此将低空参照模型取为 P_2 。高空参照模型取为 P_3 。

用模糊模型定义变动的控制对象模型如下:

$$P_1 = \alpha_2 P_2 + \alpha_3 P_3 \quad (\text{a. 25})$$

式中, $\alpha_2 + \alpha_3 = 1, \alpha_2 \geq 0, \alpha_3 \geq 0$

上式表示了二个参照模型 P_2, P_3 的参数值的内插。 P_1 为变动点中的 P_v 的线性近似模型, 表示变动的全域。 α_i 用本书图 5.17 所示的来自高度 h 的隶属函数值, 来评价 α 对变动的控制对象和参照模型的鉴定信号的应答特性的类似度。

A.3.2 模糊模型式控制器

把关于操纵的知识的概要用模糊模型的形式表示出来的是本书的式 (5.31)~式 (5.33)。由模糊控制器得到的操作量 u^* 用贡献度 V_i 在式 (5.34) 中给出。

A.3.3 直升机控制用模糊控制器的稳定性讨论

用基于模糊模型的控制器公式化得到的稳定判别条件, 来讨论直升机模糊控制器的稳定性。

(1) 关于操作模型结构的讨论这里讨论最初从操纵者提取的操作模型, 能否对随特性变动的直升机进行稳定化。首先考虑操作模型的结构。最初的操作模型为一个控制器, 控制对象直升机具有如式 (a.24) 所示的变动参数 k 。因此, 作为前述同时稳定化问题, 讨论其补偿器的结构。将式 (a.24) 的直升机传递函数进行不可约分解, 可得

$$P_v = \tilde{D}_v^{-1} \tilde{N}_v \quad (\text{a.26})$$

式中

$$D_v = \begin{bmatrix} \frac{s}{s+1} & 0 \\ 0 & \frac{s^2}{(s+2)(s+3)} \end{bmatrix} \in S$$

$$\tilde{N}_v = \begin{bmatrix} \frac{G_v}{s+1} \\ \frac{G_v}{(s+2)(s+3)} \end{bmatrix} \in S$$

$$G_v = \frac{ke_0}{s^2 + kd_3s + kd_2} \in S$$

这里 G_v 为机体的本体传递函数, 根据本体内的陀螺罗盘反馈, 对 k 的变动也是常稳定有理函数。 k 的变动范围内的任意两个控制对象为 P_i, P_j 把式 (a. 14) 的下标 1, 2 和式 (a. 14) 的下标代换成与控制对象对应的 i, j 。把由

$$R_i = \begin{bmatrix} \frac{G_j}{G_i} & 0 \\ 0 & \frac{G_j}{G_i} \end{bmatrix} \in S$$

$$R_j = \begin{bmatrix} \frac{G_i}{G_j} & 0 \\ 0 & \frac{G_i}{G_j} \end{bmatrix} \in S$$

$$X_i = \begin{bmatrix} \frac{h_1 G_j G_i - G_j s^2}{G_i (s+2)(s+3)} & 0 \\ 0 & \frac{h_0 G_j G_i - G_j s}{G_i (s+1)} \end{bmatrix} \in S$$

$$X_j = \begin{bmatrix} \frac{h_1 G_i G_j - G_i s^2}{G_j (s+2)(s+3)} & 0 \\ 0 & \frac{h_0 G_i G_j - G_i s}{G_j (s+1)} \end{bmatrix} \in S$$

$$Y_i = Y_j = 0 \in S$$

得到的 $R_i, R_j, X_i, X_j, Y_i, Y_j$ 代入式 (a. 14) 的同时稳定化条件。如果用其结果, 满足式 (a. 14) 同时稳定化可能性的条件, 可得用

$$C_i = \begin{bmatrix} h_1 & 0 \\ 0 & h_0 \end{bmatrix} \in S \quad (\text{a. 27})$$

表示的同时稳定化补偿器 C_i 。式中 h_0, h_1 为稳定有理函数。

由以上结果, 这个控制对象对参数变动范围的任意两个控制对象用式 (a. 27) 的输出反馈型控制器同时稳定化是可能的。也就是说, 由操纵者进行的操作模型 f_1 的结构, 对特性变动的控制对象进行稳定化是可能的。

参考文献

第1章 人工神经网络和模糊信号处理概述

- [1] 麻生英樹：“ニューラルネットワーク情報処理”，産業図書 (1988).
- [2] 片方善治：“知能システム工学”，海文堂 (1993).
- [3] 萩原将文：“ニューロ・ファジィ・遺伝的アルゴリズム”，産業図書 (1994).
- [4] 計測自動制御学会 (編)：“ニューロ・ファジィ・AI ハンドブック”，オーム社 (1994).
- [5] S. Haykin：“Introduction to Adaptive Filters”，Macmillan (1984).
武部 幹 (訳)：“適応フィルタ入門”，現代工学社 (1990).
- [6] 谷萩隆嗣：“デジタル信号処理の理論3—推定・適応信号処理—”，コロナ社 (1986).
- [7] F. E. Bloom, A. Lazerson and L. Hofstadter：“Brain, Mind, and Behavior”，W. H. Freeman and Company (1985).
久保田競 (監訳)：“脳の探検 (上・下)”，講談社 (1987).
- [8] 小林 繁, 熊倉鴻之助, 黒田洋一郎, 畠中 寛：“絵ときブレインサイエンス入門”，オーム社 (1987).
- [9] L. A. Zadeh：“Fuzzy sets”，Information and Control, 8, pp. 338-353 (1965).
- [10] 寺野寿郎 (監修)：“実用ファジィ制御”，電子情報通信学会 (1991).
- [11] 高木友博, 山口 亨, 菅野道夫：“概念ファジィ集合とその連想記憶による実現”，第7回ファジィシステムシンポジウム, pp. 359-364 (1991).
- [12] L. J. J. Wittgenstein：“Philosophical Investigations”，Basil Blackwell, Oxford (1953).
- [13] E. H. Mamdani and S. Assilian：“An experiment in linguistic synthesis with a fuzzy logic controller”，Int. J. Man-Machine Studies, 7, pp. 1-13 (1973).
- [14] 菅野道夫：“ファジィ制御”，日刊工業新聞社 (1988).
- [15] 廣田 薫：“ファジィ制御とセンシング技術”，第7回センシングフォーラム, pp. 7-10 (1989).
- [16] G. J. Klir and T. A. Folger：“Fuzzy Sets, Uncertainty, and Information”，Prentice-Hall (1988).
- [17] 山口 亨：“ファジィ連想記憶システム”，日本ファジィ学会誌, 5, 2, pp. 245-

- [18] 山口 亨: “カオスと連想記憶”, 日本ファジィ学会誌, 7, 3, pp. 500-511 (1995. 6).

第2章 层次型神经网络的信号处理

- [1] D. E. Rumelhart, J. L. McClelland and the PDP Research Group : “Parallel Distributed Processing”, MIT Press (1986).
甘利俊一 (監訳): “PDPモデル”, 産業図書 (1989).
- [2] T. J. Sejnowski and C. R. Rosenberg : “Parallel networks that learn to pronounce English text”, Complex Systems, 1, pp. 145-168 (1987).
- [3] R. P. Gorman and T. J. Sejnowski : “Analysis of hidden units in a layered network trained to classify sonar targets”, Neural Networks, 1, 1, pp. 75 -89 (1988).
- [4] B. Widrow et al. : “Adaptive noise cancelling : Principles and applications”, Proc. IEEE, 63, 12, pp. 1692-1716 (Dec. 1975).
- [5] J. Hertz, A. Krogh and R. G. Palmer : “Introduction to the Theory of Neural Computation”, Addison Wesley (1992).
- [6] 相吉英太郎, 吉川 厚: “ニューラルネットワークによる最適化, その疑問と課題—もう1つの接近法—”, 計測と制御, 34, 5, pp. 358-366 (1995. 5).
- [7] C. Peterson and E. Hartman : “Explorations of the mean field theory learning algorithm”, Neural Networks, 2, pp. 475-494 (1989).
- [8] S. Haykin : “Neural Network -A Comprehensive Foundation-, ” Macmillan (1994).
- [9] M. Hagiwara and A. Sato : “Theoretical derivation of momentum term in back-propagation”, IEICE Trans. Information and System, E78-D, 8, pp. 1080-1086 (Aug. 1995).
- [10] H. Sawai, A. Waibel, R. Haffner, Ivl. Miyatake and K. Shikano : “Parallel-ism, hierarchy, scaling in time-delay neural networks for spotting Japanese phonemes /CV-syllables”, Int. Joint Conf. on Neural Networks 89, II, pp. 81-88 (1989).
- [11] S. E. Fahlman : “Fast-learning variations on back-propagation : An empirical study”, 1988 Connectionist Models Summer School, pp. 38-51, Morgan Kaufmann (1989).
- [12] E. B. Baum and D. Haussler : “What size net gives valid generalization ?”, Neural Computation, 1, pp. 151-160 (1989).
- [13] D. R. Hush and B. G. Horne : “Progress in supervised neural networks”, IEEE Signal Processing Magazine, pp. 8-39 (Jan. 1993).
- [14] A. S. Weigend, D. E. Rumelhart and B. A. Huberman : “Generalization by

- weight-elimination applied to currency exchange rate prediction", Int. Joint Conf. on Neural Networks 91, I, pp. 837-841 (1991).
- [15] 私信 (D. E. Rumelhart)
 - [16] M. Hagiwara : "A simple and effective method for removal of hidden units and weights" , Neurocomputing, 6, pp. 207-218 (Mar. 1994).
 - [17] 真島憲仁, 渡邊見子, 吉村 明, 永野 俊 : "中間層素子削減のための評価関数 E-factorとその有効性" , 日本神経回路学会誌, 1, 2, pp. 64-68 (1994).
 - [18] M. I. Jordan : "Serial order : A parallel distributed processing approach" , ICS Report 8604, UC San Diego (Apr. 1986).
 - [19] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. J. Lang : "Phoneme recognition using time-delay neural networks" , IEEE Trans. Account., Speech, Signal Processing, ASSP-37, 3, pp. 328-339 (Mar. 1989).
 - [20] J. Moody and C. J. Darken : "Fast learning in networks of locally-tuned processing units" , Neural Computation, 1, pp. 281-294 (1989).
 - [21] J. Park and I. W. Sandberg : "Universal approximation using radial-basis-function networks" , Neural Computation, 3, pp. 246-257 (1991).
 - [22] M. T. Musavi, W. Ahmed, K. H. Chan, K. B. Faris and D. M. Hummels : "On the training of radial basis function classifiers" , Neural Networks, 5, pp. 595-603 (1992).
 - [23] T. Poggio and F. Girosi : "Networks for approximation and learning" , Proc. IEEE, 78, 9, pp. 1481-1497 (Sept. 1990).
 - [24] S. Lee and R. M. Kil : "A Gaussian potential function network with hierarchically self-organizing learning" , Neural Networks, 4, pp. 207-224 (1991).
 - [25] 榑原 進 : "数理学 ウェーブレット ビギナーズガイド" , 東京電機大学出版局 (1995).
 - [26] 小林邦和, 島岡豊士, 吉田信夫 : "ネットワーク最適化機能をもつウェーブレットニューラルネットワーク" , 電子情報通信学会論文誌, J77-D-II, 10, pp. 2121-2129 (1994. 10).
 - [27] T. Kohonen : "Self-Organization and Associative Memory" , Springer Verlag (1989).
 - [28] T. Kohonen : "The self-organizing map" , Proc. IEEE, 78, 9, pp. 1464-1480 (Sept. 1990).
 - [29] H. Ritter and K. Schulten : "Neural Computation and Self-Organizing Maps" , Addison Wesley (1992).
 - [30] E. McDermott and S. Katagiri : "LVQ-based shift-tolerant phoneme recognition" , IEEE Trans. Signal Processing, 39, 6, pp. 1398-1411 (June 1991).

第3章 互联型神经网络的信号处理

- [1] N. Nakano : "Associatron - A model of associative memory" , IEEE Trans. Syst., Man, and Cybern., SMC-2, 3, pp. 380-388, (May/June 1972).
- [2] 福島邦彦 : "時空間パターンの連想記憶" , 電子通信学会論文誌D, 56-D, 7, pp. 401-407 (1973.7).
- [3] J. J. Hopfield : "Neural networks and physical systems with emergent collective computational abilities" , Proc. National Academy of Sciences, USA 79, pp. 2554-2558, (1982).
- [4] Y. Hirai : "A model of human associative processor (HASP)" , IEEE Trans. Syst., Man, and Cybern., SMC-13, 5, pp. 851-857 (Sept./Oct. 1983).
- [5] B. Kosko : "Bidirectional associative memories" , IEEE Trans. Syst., Man, and Cybern., SMC-18, 1, pp. 49-60 (Jan./Feb. 1988).
- [6] M. Hagiwara : "Multidirectional associative memory" , Int. Joint Conf. on Neural Networks, I, pp. 3-6 (Jan. 1990).
- [7] D. E. Rumelhart, J. L. McClelland and the PDP Research Group : "Parallel Distributed Processing" , MIT Press (1986).
甘利俊一(監訳) : "PDPモデル" , 産業図書 (1989).
- [8] D. Amit, H. Gutfreund and H. Sompolinsky : "Information storage in neural networks with low levels of activity" , Physical Review A, 35, pp. 2293-2303 (1987).
- [9] H. Oh and S. C. Kothari : "Adaption of the relaxation method for learning in bidirectional associative memory" , IEEE Trans. Neural Networks, 5, 4, pp. 576-583 (July 1994).
- [10] M. Hattori, M. Hagiwara and M. Nakagawa : "Quick learning for bidirectional associative memory" , IEICE Trans. Information and Systems, E77-D, 4, pp. 385-392 (Apr. 1994).
- [11] Y. Takefuji : "Neural Network Parallel Computing" , Kluwer Academic Publishers (1992).
- [12] C. Peterson and E. Hartman : "Explorations of the mean field theory learning algorithm" , Neural Networks, 2, pp. 475-494 (1989).
- [13] C. Peterson : "Mean field theory neural networks for feature recognition, content addressable memory and optimization" , Connection Science, 3, 1, pp. 3-33 (1991).
- [14] E. Hartman : "A high storage capacity neural network content addressable memory" , Network, 2, pp. 315-334 (1991).

第4章 模糊性与模糊集合

- [1] L. A. Zadeh : "Fuzzy sets" . Information and Control, 8, pp. 338-353 (1965).
- [2] E. H. Mamdani : "Applications of Fuzzy algorithms for control of simple dynamic plant" , Proc. IEE, 121, 12, pp. 1585-1588 (Dec. 1974).
- [3] B. Kosko : "Newral Networks and Fuzzy Systems" , Prentice-Hall (1992).
- [4] G. J. Klir and T. A. Folger : "Fuzzy Sets, Uncertainty, and Information" , Prentice-Hall (1988).
- [5] G. J. Klir and T. A. Folger, 本多中二 (訳) : "ファジィ情報学" , 日刊工業新聞社 (1993).
- [6] L. J. J. Wittgenstein : "Philosophical Investigations" , Basil Blackwell, Oxford (1953).
- [7] 高木友博, 山口 亨, 菅野道夫 : "概念ファジィ集合とその連想記憶による実現" , 第7回ファジィシステムシンポジウム, pp. 359-362 (1991).
- [8] 菅野道夫 : "あいまい理論 (III)" , 計測と制御, 22, 5, pp. 38-42 (1983.5).

第5章 模糊推理与模糊信号处理

- [1] 菅野道夫 : "ファジィ制御" , 日刊工業新聞社 (1988).
- [2] T. Takagi and M. Sugeno : "Fuzzy identification of systems and its applications to modeling and control" , IEEE Trans. Syst., Man, and Cybern., SMC-15, 1, pp. 116-132 (Jan./Feb. 1985).
- [3] 山口 亨 : "ファジィ連想記憶システム" , 日本ファジィ学会誌, 5, 2, pp. 245-260 (1993.4).
- [4] H. Takahashi and H. Minami : "Subjective evaluation modeling using fuzzy logic and a neural network" , 3rd IFSA Congress, pp. 520-523 (1989).
- [5] 藤井輝夫, 浦 環 : "未熟なファジィ・コントローラ起動によるニューラルネット・コントローラの自動生成" , 第29回計測自動制御学会学術講演会, pp. 389-390 (1990).
- [6] 森田 温, 今井祥人, 竹垣盛一 : "神経回路型ファジィ知識モデルによる加工ノウハウの学習" , 計測自動制御学会第27回学術講演会, JS33-3, pp. 347-348 (1988).
- [7] 休 勲, 高木英行 : "神経回路網モデルによるファジィ推論の定式化" , 第4回ファジィシステムシンポジウム, pp. 55-60 (1988).
- [8] 林 陽一, 中井正人 : "ニューラルネットワークを用いたファジィプロダクションルールの自動抽出" , 第5回ファジィシステムシンポジウム, pp. 169-176 (1989).
- [9] 山口 亨, 今崎直樹, 春木和仁 : "連想記憶によるファジールールの推論・学習方式" , 電気学会論文誌 C, 110, 3, pp. 207-215 (1990.3).

- [10] 市橋秀友, 渡辺俊彦: “簡略ファジィ推論を用いたファジィモデルによる学習型制御”, 日本ファジィ学会誌, 2, 3, pp. 429-437 (1990.6).
- [11] 堀川慎一, 古橋 武, 大熊 繁, 内川嘉樹: “ニューラルネットワークによる学習型ファジィ制御器”, 計測自動制御学会論文集, 27, 2 pp. 208-215 (1991.2).
- [12] B. Kosko: “Fuzzy cognitive maps”, Int. J. Man-Machine Studies, 24, pp. 65-75 (1990).
- [13] K. Goto, T. Yamaguchi and T. Takagi: “Dynamic model for a plant using associative memory system”, Int. Conf. on Artificial Neural Networks (ICANN-92), pp. 1517-1520 (1992).
- [14] T. Yamakawa and S. Tomoda: “A Fuzzy neuron and its application to pattern recognition”, 3rd IFSA Congress, pp. 30-38 (1989).
- [15] Y. Hayashi, J. Buckley and E. Czogala: “Direct fuzzification of neural networks and fuzzified delta rule”, 2nd Int. Conf. on Fuzzy Logic & Neural Networks IIZUKA' 92, pp. 73-76 (1992).
- [16] M. Hagiwara: “Extended fuzzy cognitive maps”, IEEE Int. Conf. on Fuzzy Systems, pp. 795-801 (1992).
- [17] 石淵久生, 岡田英彦, 田中英夫: “ファジィ入出力データを用いたニューラルネットの学習”, 日本ファジィ学会誌, 4, 4, pp. 892-905 (1992.8).
- [18] 高木友博, 山口 亨, 菅野道夫: “概念ファジィ集合とその連想記憶による実現”, 第7回ファジィシステムシンポジウム, pp. 359-364 (1991).
- [19] B. Kosko: “Neural Networks and Fuzzy Systems”, Prentice-Hall (1992).
- [20] T. Yamaguchi, T. Takagi and T. Mita: “Self-organizing control using fuzzy neural networks”, Int. J. Control, 56, 2, (Aug. 1992).
- [21] 山口 亨, 関根 智, D. Montgomery, 遠藤経一: “ファジー連想推論に基づく知的インターフェースとそのコマンドスベル修正器への適用”, 電気学会論文誌 C, 113, 9, pp. 709-718 (1993. 9).
- [22] 北村新三: “ファジィルール制御系の絶対安定条件 円板定理の拡張”, 第5回ファジィシステムシンポジウム, pp. 505-508 (1989).
- [23] 北條達也, 寺野寿郎, 増井重弘: “位相面によるファジィ制御系の安定性解析”, 日本ファジィ学会誌, 4, 6, pp. 1133-1146 (1992. 12).
- [24] 長谷川貴史, 古橋 武, 内川嘉樹: “離散事象表現によるファジィ制御系の安定性解析に関する一提案”, 電気学会産業システム情報化研究会, pp. 19-28 (1995).
- [25] K. Tanaka and M. Sugeno: “Stability analysis and design of fuzzy control systems”, Fuzzy Sets and Systems, 45, 2, pp. 135-156 (1992).
- [26] 山口 亨, 栗山和信, 美多 勉: “ファジィモデルによる飛行ビーグル制御と安定性の検討”, 日本ロボット学会誌, 10, 4, pp. 501-509 (1992.8).

第6章 神经网络与模糊信号处理应用

- [1] K. Funahashi : "On the approximate realization of continuous mapping by neural networks" , Neural Networks, 2, pp. 183-192 (1989).
- [2] B. Kosko : "Fuzzy systems as universal approximators" , IEEE Int. Conf. on Fuzzy Systems, pp. 1153-1162 (1992).
- [3] 谷萩隆嗣, 高野裕昭 : "カテゴリーを組み合わせたニューラルネットワークによる顔画像認識" , 電子情報通信学会論文誌D-II, J 77-D-II, 11, pp. 2151-2159 (1994. 11).
- [4] 堀田健一, 岩田 彰, 松尾啓志, 鈴木宣夫 : "大規模ニューラルネットワークーCombNET-II-" , 電子情報通信学会論文誌 D-II, J75-D-II, 3, pp. 545-553 (1992. 3).
- [5] M. I. Jordan and R. A. Jacobs : "Hierarchical mixtures of experts and the EM algorithm" , Neural Computation, 6, pp. 181-214 (1994).
- [6] S. S. Fels and G. E. Hinton : "Glove-talk : A neural network interface between a data-glove and a speech synthesizer" , IEEE Trans. Neural Networks, 4, 1, pp. 2-8 (Jan. 1993).
- [7] 落合慶広, 鈴木英人, 篠沢一彦, 藤井雅晴, 曾根原登 : "ニューラルネットワークを用いた気象レーダ画像による降雨・降雪予測" , 電子情報通信学会技術報告, NC95-15, pp. 111-118 (1995.5).
- [8] 落合慶広, 戸田尚宏, 臼井史朗 : "重みの振動を抑制する階層型ニューラルネットワークの学習法-Kick Out法-" , 電気学会論文誌 C, 113-C, 12, pp. 1154-1162 (1993. 12).
- [9] 土佐尚子 : "音声で人間とコミュニケーションできるロボティクスアート:ニューロベビー" , 計測と制御, 34, 4, pp. 311-314 (1995. 4).
- [10] N. Tosa and R. Nakatsu : "Human-like communication character, "MIC" & feeling improvisation character, "MUSE" " , Proc. of Artificial Life V, The Fifth International Workshop on the Synthesis and Simulation of Living Systems, pp. 122-129 (June 1996).
- [11] 寺野寿郎 (監修), 国際ファジィ工学研究所(編) : "ファジィ思考による知的情報処理" , コンピュータ・エージ社 (1995).
- [12] 恒川裕史 : "ニューラルネットワークを用いた地震動の初期微動による主要動予測手法" , 第5回インテリジェント・システム・シンポジウム, pp. 107-112 (1995).
- [13] 久保 進, 中井章二, 今崎直樹 : "ファジィニューラルネット応用エレベーター群管理システム" , 東芝レビュー, 49, 9, pp. 627-630 (1994.9).
- [14] 今崎直樹, 関根 智, 愛須英之, 加納 誠, 牧野恭子, 石塚 晃 : "学習システムの構築法および記述言語" , 第35回学術講演会 SICE' 96, pp. 559-560 (1996).

- [15] 日経コンピュータ編集部：“ファジイな概念を手軽に扱う”，日経コンピュータ，398，pp. 110-111 (1996.8).
- [16] 張 育銘，森 典彦：“ニューラルネットで統合するファジイ逆推論を用いた自動車デザイン支援システム”，日本ファジイ学会誌，5，5，pp. 1233-1245 (1993. 10).
- [17] 高橋 宏：“自己帰帰移動平均モデル係数から評価知識を生成した運転性評価ファジイモデル”，日本ファジイ学会誌，5，2，pp. 330-347 (1993. 4).

第7章 人类的并行处理模型和智能信号处理

- [1] 安西祐一郎：“認知科学と人工知能”，共立出版 (1987. 11).
- [2] 久田光彦：“運動のパターンジェネレーターとは何か”，伊藤正男 (編)：“運動の中核メカニズム”，医歯薬出版 (1979).
- [3] 横山光雄：“スペクトル拡散通信システム”，科学技術出版社 (1988. 8).
- [4] M. Hattori and M. Hagiwara：“Episodic associative memory”，Neurocomputing，12，pp. 1-18 (1996).
- [5] 合原一幸 (編著)：“ニューロ・ファジイ・カオス”，オーム社 (1993).
- [6] 合原一幸 (編著)：“ニューラルシステムにおけるカオス”，東京電機大学出版局 (1993).
- [7] P. Davis, 奈良重俊：“カオスと情報処理機能”，合原一幸 (編)“カオス”，サイエンス社 (1990).
- [8] 谷 淳：“カオスの最急降下法を適用したニューラルネットワークにおける学習および記憶想起の動特性について”，電子情報通信学会論文誌A，J74-A，8，pp. 1208-1215 (1991.8).
- [9] 長名優子，服部元信，萩原将文：“カオス双方向連想メモリ”，電気学会論文誌C，116-C，7，pp. 741-747 (1996. 7).
- [10] 山口 亨，吉原道宏，石川欽彦：“連想記憶による知的ノシバーバルインタフェース”，第5回インテリジェント・システム・シンポジウム論文集，pp. 385-390 (1993).
- [11] J. Rasmussen：“Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models”，IEEE Trans. Syst., Man, and Cybern., SMC-13, 3, pp. 257-266 (May/June 1983).
- [12] T. Yamaguchi, M. Yoshihara et al.：“Japanese sign language recognition system using information infrastructure”，The Fourth IEEE Int Conf. on Fuzzy Systems and the Second Int. Fuzzy Engineering Symposium, 5, pp. 65-66 (1995).
- [13] 秋田幸治，小俣寿之，橋本智己，山口 亨：“動作指示による階層的学習機能を持つ自律移動ロボット”，第5回インテリジェント・システム・シンポジウム

論文集, pp. 91-96 (1995).

- [14] 佐藤友彦, 牛田博英, 山口 亨, 井村 敦, 高木友彦: “ファジィ連想記憶システムにおけるカオスの記憶探索”, 日本ファジィ学会誌, 6, 4, pp. 765-774 (1994. 8).
- [15] 山口 亨: “カオスと連想記憶”, 日本ファジィ学会誌, 7, 3, pp. 500-511(1995. 6).
- [16] 山口 亨, 吉原道広, 石川欽彦, 小俣寿之: “連想記憶を用いたインタラクティブ手話インタフェースの試作”, 日本ファジィ学会誌, 8, 5, pp. 901-907 (1996. 10).
- [17] 津田一郎: “カオスの脳観”, サイエンス社 (1990).
- [18] 奈良重俊, P. Davis: “カオスによるパターン発生とその適応制御”, システム/制御/情報, 37, 11, pp. 654-660 (1993. 11).
- [19] 國藤 進: “発想支援システムの研究開発動向とその課題”, 人工知能学会誌, 8, 5, pp. 552-559 (1993. 10).
- [20] 高木友博, 山口 亨, 菅野道夫: “概念ファジィ集合とその連想記憶による実現”, 第7回ファジィシステムシンポジウム講演論文集, pp. 359-362 (1991).
- [21] 馬場敬信, 吉永 努: “並列オブジェクト指向トータルアーキテクチャA-NETにおける言語とアーキテクチャの統合”, 電子情報通信学会論文誌D-I, J75-D-I, 8, pp. 563-574 (1992. 8).

附 录 模糊系统与稳定性

- [1] 山口 亨, 栗山和信, 美多 勉: “ファジィモデルによる飛行ビーグル制御と安定性の検討”, 日本ロボット学会誌, 10, 4, pp. 501-509 (1992. 8).
- [2] G. Obinata and J. B. Moore: “Characterization of controllers in simultaneous stabilization”, System & Control Letter, 10, 5, pp. 333-340 (1988).
- [3] M. Vidyasagar: “Control System Synthesis”, MIT Press (1985).

索 引

【B】

玻尔兹曼机 48

【C】

层次型神经网络 6
层次型神经网络 8

长期记忆 142
程序性记忆 142

【D】

短期记忆 142

多代理机系统 17

【D】

反向传播 6

仿射松弛法 54

【G】

概念模糊集合 12
感知器 6

关键输入 148

【H】

核函数 36
互联型神经网络 8
回声消除器 4

混沌神经元 147
混沌双向联想存储 148
霍普菲尔德网络 48

【J】

径向基函数 20

【K】

可视化 41

可视神经元 61

【L】

连 贯 148

联想存储 48

联想存储模型 48

邻域函数 42

【M】

模糊传感器 14

模糊规则 80

模糊集合 12

模糊控制 13

模糊理论 2

模糊推理 80

模糊信号处理 14

【N】

能量函数 49

【P】

平衡状态 54

平均场理论 59

【Q】

情节记忆 142

情节联想存储 143

群管理系统 128

【R】

人工智能 6

【S】

神经网络 2

声明记忆 142

胜出神经元 42

时空间模式联想存储模型 48

双弯曲函数 8

【T】

凸函数 70

退火 59

【W】

稳定性 49

无教师学习 8

【X】

相互回想 48

小波神经网络 40

协方差学习规则 50

学习向量量化 20

【Y】

延迟线滤波器 21
意思记忆 142
隐含神经元 61

隐式马尔可夫模型 36
有教师学习 8

【Z】

智能 16
智能层次模型 152
智能工业系统 16
智能信号处理 14
自回想 48
自适应信号处理 4
自适应预测器 5

自组织化 41
自组织映射 20
组合最优化问题 49
最大成长率 29
最速下降法 21
最优化 48

Associatron 48
BAM 48
BAM 用仿射松弛学习算法 53
HASP 48
HMM 36
Jordan 网络 34
Kullback 的信息量 62

K平均法 38
LMS 算法 5
MAM 48
Melscale 方法 36
MFT 学习算法 59
PRLAB 53
VC 30

“数字信号处理参考教材系列”序

近年来，随着数字技术的惊人发展，以前用模拟技术进行处理或者以往根本无法进行数字处理的问题，都可以进行数字处理了。因此，数字技术越来越广泛地应用于诸多领域，而且这些领域对数字技术的要求也变得越来越高的。

最近对电气、电子、信息、通信等领域进行的大规模市场调查表明，很多企业以及研究机构都对数字信号处理技术非常重视，他们在调查问卷的表格中，把数字信号处理填在了“必要性”和“重要性”一栏的首位。从这一社会现象也可以看出，数字信号处理是当今社会急需发展的学科领域之一。

鉴于这种状况，我们以供从事数字信号处理或者准备学习数字信号处理的社会各界人士参考阅读为目的，从更广泛的角度对数字信号处理这一学科进行归纳整理，编写了这套系列书。

本系列书包括以下各册：

1. 数字信号处理基础理论
2. 数字滤波器与信号处理
3. 语音与图像的数字信号处理
4. 快速算法与并行信号处理
5. 卡尔曼滤波器与自适应信号处理
6. ARMA 系统与数字信号处理
7. VLSI 与数字信号处理
8. 信息通信与数字信号处理

9. 人工神经网络与模糊信号处理

10. 多媒体与数字信号处理

上述各册中,第 1 至第 3 为基础部分,以大学三、四年级本科生为读者对象;第 4 至第 6 为比基础部分内容较深的提高部分,以研究生或者具有同等学历的科研人员及技术人员为读者对象;第 7 至第 10 为应用部分,以大学或研究机构的研究人员为主要读者对象,亦可供有一定基础知识的社会各界人士参考阅读。

也就是说,读者可根据自己的兴趣和所掌握的知识基础,有选择地阅读本系列书中的内容。比如,从基础知识开始学习数字信号处理的读者,可选择基础部分的内容;如果已具备了一定的基础知识,则可选择提高部分或者应用部分。从基础知识开始学习的,可按基础部分→提高部分→应用部分的顺序,或者按基础部分→应用部分→提高部分的顺序,根据自己的兴趣有选择地阅读。

本系列的执笔者均为目前仍活跃在相关领域第一线的专家、学者,因而编者有理由相信本系列书能够满足不同层次读者的需求。

另外,考虑到数字信号处理理论及应用技术的迅速发展,今后我们会根据情况及时补充新内容,使本系列书不断充实和完善。

最后,时值本系列书出版之际,谨向对本系列书的出版提供多方帮助的 CORONA 社的各位表示衷心的感谢。

“数字信号处理参考教材系列”策划兼主编

谷 菽隆嗣

前 言

人工神经网络和模糊理论，是在对人类及众多生物的适应功能、识别功能、学习功能和判断功能等进行深入研究的基础上提出来的，对于实现高智能系统，进行智能信号处理是非常重要的。

本书首先概要介绍应用神经网络和应用模糊理论的信号处理技术。然后介绍具有代表性的层次型及互联型神经网络，并说明其信号处理方法。进而对模糊性和模糊集合、模糊推理和模糊信号处理等进行阐述。

另外，书中还列举了神经网络和模糊信号处理的各种应用实例。最后介绍了今后以高智能系统为目标的并行处理模型和智能信号处理。

第 1 章，介绍神经网络与模糊信号处理的概要，并对其特征及发展前景予以叙述。

第 2 章，介绍层次型神经网络及应用层次型神经网络的信号处理。众所周知，反向传播对层次型神经网络的学习算法很有效，因此先详细说明反向传播，然后叙述学习的快速化和网络结构的最优化。接下来介绍径向基函数网络，并举出具有代表性的学习算法。最后阐述了自组织化映射及其学习算法。

第 3 章，叙述互联型神经网络和应用互联型神经网络的信号处理。互联型神经网络，可以分为联想存储模型和用于模式识别及优化的神经网络。联想存储模型，由于与人类的

记忆模式相类似，今后将会得到进一步发展。另外，作为以最优化为目的的网络，还将叙述玻尔兹曼机及其学习算法。

第4章，叙述模糊性和模糊集合。首先介绍模糊性和模糊集合的表示，模糊集合的基本运算。然后，详细说明由其他语言形成的模糊集合把具有模糊性意思的语言进行分散表示的概念模糊集合。从信号处理的观点看，概念模糊集合也是重要的概念。

第5章，介绍模糊推理与模糊信号处理。首先介绍模糊信号处理中的模糊推理，然后给出几个关于应用神经网络的模糊推理以及模糊信号处理和模糊控制的例子予以解释。

第6章，列举神经网络和模糊信号处理的几种应用实例。介绍神经网络在模式识别、气象预测和感性信息处理等诸多领域的应用，以及模糊信号处理在建筑领域及交通系统、辅助设计和汽车等领域的应用。

第7章，介绍人类的并行处理模型与智能信号处理。首先说明用神经网络的联想存储智能信号处理。然后举例说明联想存储的模糊联想处理。最后对新型智能信号处理进行了介绍。

本书以高性能智能系统的发展为目标，介绍了神经网络和模糊信号处理。由于智能信号处理领域具有很大的发展前景，因此本书如能对此有所贡献，作者将深感荣幸。

编 著 者 谷 萩 隆 嗣

编著者简历

谷 萩 隆 嗣 (第 6 章)

- 1966年 东京工业大学工学部电子工程专业毕业
- 1971年 东京工业大学研究生院工学研究科电子工程专业毕业, 获工学博士学位
- 1971年 千叶大学讲师(工学部电子工程专业)
- 1974年 千叶大学副教授(工学部电子工程专业)
- 1984年 千叶大学教授(工学部电子工程专业)
- 1989年~现在 千叶大学教授(工学部信息工程专业)

萩 原 将 文 (第 1~3 章, 第 6 章, 第 7 章)

- 1982年 庆应义塾大学工学部电气工程专业毕业
- 1987年 庆应义塾大学研究生院工学研究科电气工程专业毕业, 获工学博士学位
- 1987年 庆应义塾大学助教(理工学部电气工程专业)
- 1990年 庆应义塾大学专职讲师(理工学部电气工程专业)
- 1995年~现在 庆应义塾大学副教授

山 口 亨 (第 1 章, 第 4~6 章, 第 7 章, 附录)

- 1979年 千叶大学工学部电气工程专业毕业
- 1981年 千叶大学研究生院工学研究科电气工程专业硕士毕业
- 1981年~1993年 (株) 东芝公司 工作
- (1989年~1992年 派赴国际模糊工程研究所)
- 1992年 获工学博士学位(千叶大学)
- 1993年~现在 宇都宫大学副教授

